ISSN 2224-087X

ЕЛЕКТРОНІКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ELECTRONICS AND INFORMATION TECHNOLOGIES

Збірник наукових праць

Випуск 32

2025

ELECTRONICS AND INFORMATION TECHNOLOGIES

ЕЛЕКТРОНІКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Issue 32

Scientific journal

Published 4 issue per year

Published since 1966

Випуск 32

Збірник наукових праць

Виходить 4 рази на рік

Видається з 1966 р.

Ivan Franko National University of Lviv

Львівський національний університет імені Івана Франка

2025

ЗАСНОВНИК: ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Друкується за ухвалою Вченої Ради
Львівського національного університету
імені Івана Франка
протокол №1/12 від 23.12.2025 р.

Реєстрація суб'єкта у сфері друкованих медіа:
Рішення Національної ради України з питань
телебачення і радіомовлення № 1877 від
30.05.2024 р. Ідентифікатор медіа R30-04912

У 1966–2010 рр. збірник виходив під назвою «**Теоретична електротехніка»**

Збірник «Електроніка та інформаційні технології» містить оригінальні результати досліджень з електронного матеріалознавства, моделювання фізичних явищ, процесів і систем електроніки, обробки сигналів і зображень, інформаційних технологій.

“Electronics and information technologies” journal contains original research results on electronics material science, modelling of physical phenomena, processes and electronic systems, signal and image processing and information technologies.

Редактори

Проф. *І. Карбовник* – головний співредактор
Проф. *І. Бордун* – головний співредактор
Проф. *О. Крупич* – відповідальний редактор
С.н.с. *Я. Шмигельський* – відповідальний секретар

Редакційна колегія

д-р техн. наук, проф. *О. Андрейків*
д-р габіл., проф. *Б. Андрієвський*
д-р фіз.-мат. наук, проф. *І. Болеста*
канд. фіз.-мат. наук, доц. *С. Вельгош*
д-р фіз.-мат. наук, проф. *П. Венгерський*
д-р техн. наук, проф. *Р. Воробель*
д-р фіз.-мат. наук, проф. *Р. Головчак*
д-р техн. наук, д-р габіл., проф. *Ф. Іващишин*
д-р фіз.-мат. наук, проф. *О. Кушнір*
д-р фіз.-мат. наук, проф. *А. Лучечко*
д-р техн. наук, проф. *Л. Муравський*
д-р техн. наук, проф. *М. Назаркевич*
д-р фіз.-мат. наук, проф. *І. Оленич*
д-р фіз.-мат. наук, проф. *Б. Павлик*
д-р техн. наук, доц. *Б. Павлишенко*
д-р техн. наук, проф. *С. Рендзіняк*
д-р фіз.-мат. наук, проф. *Б. Русин*
д-р фіз.-мат. наук, проф. *М. Притула*
д-р габіл., проф. *Ц. Славінські*
канд. фіз.-мат. наук, доц. *Ю. Фургала*
д-р габіл., проф. *Б. Ціж*,
д-р габіл., проф. *Бушта Шахрауі*
д-р фіз.-мат. наук, проф. *Г. Шинкаренко*
канд. фіз.-мат. наук, доц. *Р. Шувар*
д-р фіз.-мат. наук, проф. *І. Яворський*

Editors

Prof. *I. Karbovnyk* – Chief Co-Editor
Prof. *I. Bordun* – Chief Co-Editor
Prof. *O. Krupych* – Managing Editor
Sen. Res. *Ya. Shmygelsky* – Technical Editor

Editorial Board

O. Andreykiv, Dr. Sc., Prof.
B. Andrievsky, Dr. Habil., Prof.
I. Bolesta, Dr. Sc., Prof.
S. Velgosh, PhD, Assoc. Prof.
P. Vengersky, Dr. Sc., Prof.
R. Vorobel, Dr. Sc., Prof.
R. Holovchak, Dr. Sc., Prof.
F. Ivaschyshyn, Dr. Sc., Dr. Habil., Prof.
O. Kushnir, Dr. Sc., Prof.
A. Luchechko, Dr. Sc., Prof.
L. Muravsky, Dr. Sc., Prof.
M. Nazarkevych, Dr. Sc., Prof.
I. Olenych, Dr. Sc., Prof.
B. Pavlyk, Dr. Sc., Prof.
B. Pavlyshenko, Dr. Sc., Assoc. Prof.
S. Rendzinyak, Dr. Sc., Prof.
B. Rusyn, Dr. Sc., Prof.
M. Prytula, Dr. Sc., Prof.
C. Slawinski, Dr. Habil., Prof.
Yu. Furgala, PhD, Assoc. Prof.
B. Tsizh, Dr. Habil., Prof.
Bouchta SAHRAOUI, Dr. Habil., Prof.
G. Shynkarenko, Dr. Sc., Prof.
R. Shuvar, PhD, Assoc. Prof.
I. Yavorskyy, Dr. Sc., Dr. Habil., Prof.

Адреса редакційної колегії:
Львівський національний університет імені Івана Франка, факультет електроніки та інформаційних технологій, вул. Ген. М. Тарнавського, 107, 79017, Львів, Україна
тел. (+38) (093) 864-01-19

Editorial office address:
Ivan Franko National University of L'viv,
Faculty of Electronics and Computer Technologies
107 Tarnavsky St., UA–79017,
Lviv, Ukraine
tel. (+38) (093) 864-01-19

e-mail: elit@lnu.edu.ua
web-сайт: http://publications.lnu.edu.ua/collections/index.php/electronics/index

АДРЕСА РЕДАКЦІЇ, ВИДАВЦЯ І ВИГОТОВЛЮВАЧА:

Львівський національний університет імені Івана Франка
вул. Університетська, 1, 79000 Львів, Україна

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців, виготівників і розповсюджувачів видавничої продукції. Серія ДК № 3059 від 13.12.2007 р.

© Львівський національний університет імені Івана Франка, 2025

CONTENTS

INFORMATION SYSTEMS AND TECHNOLOGIES

MATERIALS FOR ELECTRONIC ENGINEERING

ЗМІСТ

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

МАТЕРІАЛИ ЕЛЕКТРОННОЇ ТЕХНІКИ

ISSN 2224-088X (print) • ISSN 2224-0888 (on-line)

Electronics and Information Technologies, 2025, 32, 5–18
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.9

GRAPH VISUALIZATION OF TRACEROUTE UTILITY RESULTS FOR COMPUTER NETWORK ANALYSIS

***Ivan Danych**, Kvitoslava Obelovska, Zoreslava Shpak*
Lviv Polytechnic National University
12 Stepan Bandera St., Lviv 79013, Ukraine

Danych, I.M., Obelovska, K.M., Shpak Z.Y. (2025). Graph Visualization of Traceroute Utility Results for Computer Network Analysis. *Electronics and Information Technologies*, 32, 5–18 https://doi.org/10.30970/eli.32.1

ABSTRACT

Background. Monitoring destination reachability, path discovery, and latency analysis is essential for network performance, reliability, and security. The traceroute utility is widely used for path diagnostics, but its text-based output limits the ability to perform complex analysis of network capabilities. ICMP traffic filtering, rate limiting, and load balancing cause missing hops in route displays, non-monotonic delays, and artifacts that are difficult to detect in tabular form. Existing visualization methods rely on geolocation databases and emphasize geographical aspects rather than network topology. Therefore, developing a geolocation-independent graph-based visualization tool for traceroute is relevant.

Materials and Methods. The proposed graph-based approach converts single and multiple traceroute results into a unified weighted directed graph, where vertices represent IP-defined nodes. Unknown nodes are handled using two strategies: explicit labeling or skipping with a direct connection between known nodes. The following edge-weighting criteria are introduced: occurrence frequency, used to identify critical links, and average round-trip time (RTT), used to detect latency segments. Graph visualization is performed using the Fruchterman-Reingold algorithm. The system is implemented in Java with JavaFX, Spring Core, and GraphStream, supports both archived results import and real-time tracing via a multithreaded producer-consumer model, ensuring graph integrity.

Results and Discussion. Combining multiple traceroutes into a single graph enables quick identification of shared segments and critical nodes. Systematic RTT non-monotonicity, likely caused by rate limiting, supports using absolute rather than incremental RTT values. In the experiments conducted, the multithreaded mode accelerates graph construction fourfold with seven threads, but it also increases the number of timeouts. Using five threads achieves an optimal balance between speed and data quality.

Conclusion. The proposed approach improves network analysis efficiency by constructing a geolocation-independent graph for diagnostics. The developed system reduces cognitive load compared to tabular logs analysis and speeds up detection of bottlenecks, common segments, critical nodes, and routing anomalies.

Keywords: traceroute, graph visualization, network topology, network monitoring, weighted directed graphs

INTRODUCTION

Monitoring the routes to destination hosts and analyzing their timing characteristics are key tasks for ensuring the performance, reliability, and security of network

© 2025 Ivan Danych et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

infrastructure. The traceroute utility is among the most widely used diagnostic tools for tracking the sequence of intermediate nodes along the path from source to destination [1]. However, as the number of traceroutes and the complexity of routes increase, the text-based representation of traceroute results becomes cumbersome, complicating the identification of potential bottlenecks, shared segments, critical nodes, and network behavior anomalies.

Furthermore, the results are significantly influenced by various technical factors of modern network infrastructure, such as Internet Control Message Protocol (ICMP) traffic filtering by firewalls [2], response rate limiting [3], lower processing priority of ICMP packets compared to main application traffic [4], as well as load balancing mechanisms and multipath routing [5]. These factors lead to omissions in the sequence of displayed nodes, non-monotonic RTT (round-trip time) values, and the appearance of artifacts such as loops and diamonds in the routes, which greatly complicate the interpretation of even single traceroutes [5]. All these issues are difficult to detect in a text-based output, as it requires focused analysis to establish relationships between different traceroute results.

Existing solutions for visualizing traceroute results primarily employ geographic approaches that display routing information on world maps or three-dimensional globes. Among such tools are Open Visual Traceroute for local use and various web-based services for online tracing [6-9]. Specialized systems such as VisTracer are designed to detect routing anomalies [10]. Although geographic visualization is intuitive, it has several drawbacks: dependence on external geolocation databases with potential gaps and inaccuracies, and limited functionality in cases where geographic information is unavailable. Moreover, most of the reviewed tools do not support loading archived traceroute results or simultaneous visualization of multiple routes.

This paper proposes a graph-based visualization approach for analyzing traceroute results in network analysis tasks. Single and multiple traceroute runs are transformed into a unified weighted directed graph, where vertices correspond to unique IP addresses and edges represent sequential transitions between nodes. To handle incomplete data, two strategies for processing unknown nodes are implemented: explicit representation of such nodes with appropriate labeling, or skipping them with a direct connection between adjacent known nodes. Two schemes of edge-weighting are introduced: edge occurrence frequency across multiple traceroutes to identify critical links, and RTT-based statistics (average value) to detect segments with increased latency. Visualization is performed using the classical force-directed Fruchterman-Reingold algorithm [11]. To accelerate traceroute execution, a multithreaded approach is supported. It was found that aggressive parallelism may amplify artifacts related to rate-limiting mechanisms and ICMP processing policies.

The practical usefulness of the proposed approach is demonstrated through examples of single and multiple traceroutes: combining them into a weighted graph simplifies the detection of potential bottlenecks, shared network segments, enables the localization of critical nodes, and allows for rapid identification of anomalies. The RTT-oriented weighting highlights segments with increased latency, while the frequency-based weighting emphasizes the “backbone” of routing paths. Additionally, this approach presents pedagogical value, allowing students to explore network routes and understand routing dynamics through interactive visualization. Combined with simulation-based educational methodologies explored in our previous work [12], this creates opportunities for comprehensive computer network education platforms.

Research aim: to improve the efficiency of network analysis based on the results of the traceroute utility by constructing a weighted, geolocation-independent graph suitable for broad application in diagnostics and monitoring across various practical domains.

MATERIALS AND METHODS

This study proposes an approach for converting the textual output of the traceroute utility into weighted directed graphs to enable geolocation-independent analysis of routes within a given network topology. The approach includes a data parsing and processing algorithm, two strategies for handling unknown nodes, and two edge-weighting models – RTT-oriented and frequency-based.

Principle of operation of the traceroute utility

The traceroute utility is a fundamental and widely used diagnostic tool for analyzing network paths between a source node and a destination node. Along such a path, in addition to the source and destination, a sequence of intermediate nodes is identified, along with the timing indicators for reaching each node [13]. The operation of the utility is based on the sequential manipulation of the Time To Live (TTL) field in the IP packet header, as defined in RFC 792 [14] and RFC 1812 [15]. The utility sends packets with incrementally increasing TTL values (starting from 1). Each intermediate router decreases the TTL field by one, and when the value reaches zero (TTL = 0), it sends an ICMP “Time Exceeded” message. For each node, typically three probe measurements are performed, each reporting the round-trip time (RTT), which allows for the estimation of latency and connection stability. In Windows, the utility by default limits tracing to 30 hops, while the /d parameter disables DNS resolution to speed up the process [13].

The standard output of the traceroute utility is presented in a tabular format, where each row corresponds to a single node along the route (**Fig. 1**). The structure includes: the sequential hop number, three RTT values in milliseconds (or the "*" symbol that indicates a timeout when no response is received within the waiting period, typically 5 seconds), the IP address, and optionally the domain name of the corresponding router. This format allows for assessing the path length using the number of hops to the destination, observing the list of networks through which the packet traversed, and analyzing their sequence. A more detailed examination enables identifying segments with increased latency and common path sections when comparing multiple traceroutes.

```
C:\Users\dynyc\OneDrive\Desktop>tracert -d lpnu.ua

Tracing route to lpnu.ua [178.212.110.31]
over a maximum of 30 hops:

  1     3 ms     2 ms     3 ms  192.168.0.1
  2    14 ms     6 ms     6 ms  100.65.0.1
  3     5 ms     4 ms     5 ms  193.93.218.124
  4     *        *        *     Request timed out.
  5    11 ms    13 ms    11 ms  184.104.202.190
  6    14 ms    11 ms    21 ms  178.212.110.31

Trace complete.
```

Fig. 1. Example of the standard output of the Tracert utility in the Windows operating system.

The accuracy and completeness of traceroute results are significantly affected by the technical characteristics of modern network infrastructure, which complicate data interpretation. The main influencing factors include:

- ICMP traffic filtering. Many routers and firewalls block or ignore ICMP packets for security reasons, resulting in “* * *” entries in the output instead of IP addresses and RTT values [2]. This can conceal actual intermediate devices along the route.

- Response rate limiting. Modern network equipment may employ mechanisms that limit the number of ICMP replies, potentially leading to false timeouts or non-monotonic RTT values even under normal router operation [3].
- Traffic prioritization. ICMP packets may be processed with a lower priority compared to primary application traffic (TCP/UDP), which can distort measured RTT values and may not accurately reflect the real latency experienced by user applications [4].
- Load balancing. In complex networks, traffic can be distributed across multiple paths, resulting in topological artifacts such as loops, diamonds, or parallel branches in traceroute results [5].

Conversion of traceroute textual results into a graph structure

The purpose of this transformation is to convert the sequential textual output of the traceroute utility into a directed graph $G = (V, E)$, where the vertices (V) represent unique IP addresses of network devices, and the edges (E) correspond to transitions between consecutive nodes along the route.

The process of converting a single traceroute $T = \{node_1, node_2, \dots, node_n\}$ into a graph structure is carried out according to the following algorithm:

1. Source initialization. The initial node is set as prev = "local", representing a virtual vertex corresponding to the traceroute source (the user's local machine).
2. Data preprocessing. The parser processes the input text, discarding service lines such as "Tracing route to…", "over a maximum of 30 hops", and "Trace complete". For each hop line, the following parameters are extracted: hop number, three RTT samples, and the IP address.
3. Node processing.
   a. RTT parsing: three RTT values are extracted; if at least one measurement is available, the average RTT for the node is computed.
   b. Node identification: the current node identifier is determined as node = IP (if known) or node = Unknown (if not known).
   c. Graph structure update: node is added to the vertex set V (if not already present); a directed edge (prev → node) is created and added to the edge set E.
   d. State update: prev = node is set for the next iteration.
4. Completion. The target host is recorded as the final vertex of the route.

A particular challenge is posed by nodes without ICMP responses (marked as "* * *"). The absence of an ICMP reply should not be automatically interpreted as a physical break in the path, since network devices may intentionally ignore ICMP packets. Two alternative processing strategies for such cases, selectable through configuration, are implemented.

The first approach uses explicit representation of unknown nodes. Each unknown hop is treated as a separate vertex with a specific label and color – yellow and marked as "Unknown" (**Fig. 2a**). This approach preserves route continuity and explicitly visualizes the uncertainty of a node's identity. However, it also increases the graph density, especially considering that during multiple traceroutes, several unknown vertices may physically correspond to the same node.

The second approach skips unknown nodes. An unknown node is skipped, and adjacent known nodes are connected directly (**Fig. 2b**). The presence of an unknown segment is recorded in the edge metadata. This approach improves the readability of the graph by making it more compact. At the same time, the actual route length becomes compressed, introducing a risk of creating physically impossible direct connections.

The developed system enables the transformation of a set of traceroutes into a single unified graph. In this case, the algorithm is similar to the one described above but is applied sequentially to multiple individual traceroutes. The key feature of the merging process lies in the aggregation of corresponding metrics: if the processed traceroute contains an edge

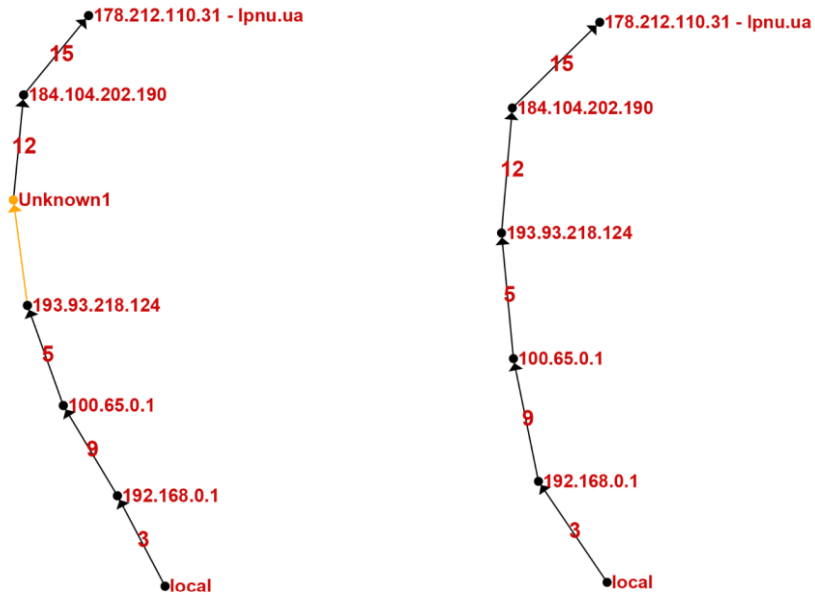


Fig. 2. Example of graph construction for the traceroute shown in Fig. 1: a) explicit representation of an unknown node; b) omission of the unknown node.

that already exists in the graph, the new RTT measurements are aggregated with the existing ones (updating the average value across all traceroutes), thereby providing more accurate data for edge weighting.

To transform the topological graph into an analytical tool for network monitoring, an edge-weighting system was developed that enables quantitative evaluation of various network aspects. The weighting models provide contextual insight for identifying critical nodes, bottlenecks, and anomalous routing patterns that cannot be detected through simple connectivity analysis of the graph. Two basic weighting models are applied in this study (**Fig. 3**).

RTT-based model. For each edge, the weight is determined based on the set of all RTT measurements. The primary estimate is the average value across corresponding samples, which highlights route segments with increased latency (**Fig. 3a**). A fundamentally important feature is the use of the absolute RTT value to the target node

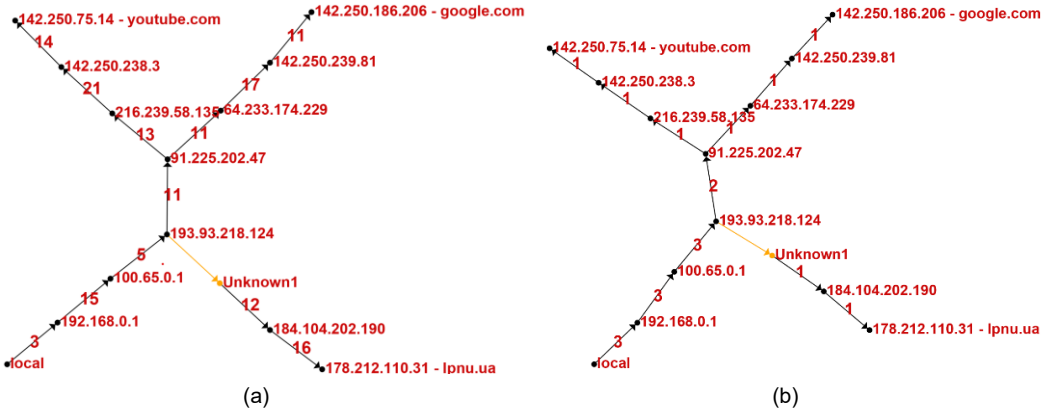


Fig. 3. Example of traceroute visualization to three resources: lpnu.ua, google.com, youtube.com. a) RTT-based model; b) Frequency-based model.

instead of the incremental delay between consecutive nodes. This decision is motivated by practical observations: due to rate-limiting mechanisms and temporary router overloads, it is common for a node at position N to exhibit a higher delay than the subsequent node at position *N* + 1. Using incremental values in such cases would result in negative weights or distortion of the actual network quality representation.

*Frequency-based mode*l. For each edge, the weight equals the number of traceroutes in which that edge appears (**Fig. 3b**). This weight is interpreted as an indicator of the segment's commonality and helps highlight the key connections (the "backbone") within the set of routes. For example, in the case of three independent traceroutes, edges shared by all routes would have a weight of 3, indicating their role as a common provider segment.

In the future, the proposed weighting approach for nodes and edges is expected to be extended. Alternative metrics – such as response stability, delay variance, and packet loss frequency – could provide a more comprehensive real-time network analysis.

Software implementation

For the practical implementation of the described algorithms for transforming and visualizing traceroute results, a specialized module was developed to extend the functionality of an existing graph visualization system created in previous research [16]. The base system provides a platform for visual analysis of graph structures with support for various visualization algorithms and interactive exploration tools, which allowed us to focus on the specific aspects of processing textual traceroute data.

The developed module supports two main operating modes:

- Text data parsing mode – designed for analyzing previously collected traceroute results. The module allows importing text files and applies the developed parser to structurally extract data on nodes, RTT values, and IP addresses, after which it executes the selected strategy for handling unknown nodes. The user can dynamically choose the weighting model (RTT-oriented or frequency-based) and adjust visual display parameters.
- The online tracing mode provides interactive execution of the traceroute utility with graph construction in near real time. The user specifies a list of target resources through the graphical interface, after which the system automatically initiates multithreaded tracerouting and gradually integrates the results into the visualization as they are received.

The online tracing mode is implemented using a multithreaded approach, where the traceroute, utility is executed in parallel at the operating system level. The system simultaneously launches multiple tracert processes for different target resources.

To achieve this, the producer–consumer pattern with a BlockingQueue is used. The results of each traceroute (tracing threads) are placed into a dedicated queue, which is processed by a single consumer thread. Formally, the software implementation can be described as follows:

$$Producers: P_1, P_2, \dots, P_n \xrightarrow{Queue} Consumer\ (Sequential) \quad (1)$$

where P_n represents the *n*-th tracing thread, and the arrow denotes the asynchronous data transfer through the queue followed by sequential processing.

This design is motivated by the need to ensure the integrity of the graph structure: since the graph is a shared resource modified when adding new vertices and edges, parallel update operations may lead to race conditions, data loss, or structural inconsistencies. Sequential queue processing guarantees the atomicity of operations and the consistency of the resulting graph.

The development of the online mode was driven by two main factors. First, there is a lack of large publicly available datasets of traceroute results suitable for research and validation of the proposed solution. Second, advancing the concept of real-time network analysis requires periodic data updates, enabling the tool to function as an interactive and efficient monitoring system.

The number of parallel threads n can be configured in the system's configuration file. During practical testing, it was found that beyond a certain threshold of threads, traceroute operations begin to interfere with each other, leading to an increase in nodes with timeouts. This behavior is likely related to rate-limiting mechanisms at the network equipment level and ICMP traffic handling policies.

The software implementation is built on the Java platform, using JavaFX for the graphical user interface, Spring Core for configuration and service management, and GraphStream for working with graph structures. This combination of libraries provides flexibility, modularity, and extensibility of the system.

RESULTS AND DISCUSSION

This section presents the experimental results obtained for the developed system. The study was organized in two main stages: first, the analysis of archived traceroutes to demonstrate the system's basic functionality and its ability to identify shared segments and critical nodes; and second, a series of experiments evaluating the performance of the multithreaded mode and identifying limitations related to rate-limiting mechanisms.

Analysis of multiple traceroutes based on the results of the traceroute utility

To demonstrate the effectiveness of the graph-based approach to network topology analysis, a comprehensive evaluation was conducted using multiple traceroutes to six different resources: google.com, youtube.com, student.lpnu.ua, lpnu.ua, chatgpt.com, and deepl.com, based on previously collected textual traceroute results. The results were integrated into a single weighted directed graph using an RTT-oriented weighting model, where numerical labels on the edges represent the average RTT values in milliseconds, and unknown nodes are highlighted in yellow (**Fig. 4**).

A detailed analysis of individual graph branches reveals an important characteristic of RTT measurements – a systematic violation of monotonic delay growth along the route. On the branch to google.com, the delay to node 142.250.239.81 is 17 ms, while the next node, 142.250.186.206, shows an RTT of only 11 ms. A similar situation was observed for the route to youtube.com, where the intermediate node 142.250.238.3 has an RTT of 21 ms, and the final node 142.250.75.14 shows only 14 ms. Such anomalies may result from rate-limiting mechanisms on intermediate routers [3] or temporary overloads on certain nodes. These observations confirm the correctness of using absolute RTT values to the target node instead of increments between consecutive hops, since the latter could produce negative weights and distort the actual network latency pattern.

Combining six traceroutes into a single graph demonstrates the key advantage of the visual approach: the shared provider segment local → 192.168.0.1 → 100.65.0.1 → 193.93.218.124 can be visually identified instantly, whereas in a text format this would require careful comparison of six separate result tables, each containing five to eight rows. This segment represents the backbone of the local network infrastructure. It is also worth noting that in this format, it is easy to interpret latency times to different resources – in our case, all of them exhibit a similarly low delay of 11-16 ms. Overall, this approach greatly facilitates the detection of potential bottlenecks.

Graph-based visualization effectively reveals more complex network relationships. The routes to student.lpnu.ua and lpnu.ua branch after the node 193.93.218.124. Interestingly, both routes contain unknown nodes in similar positions, and the node

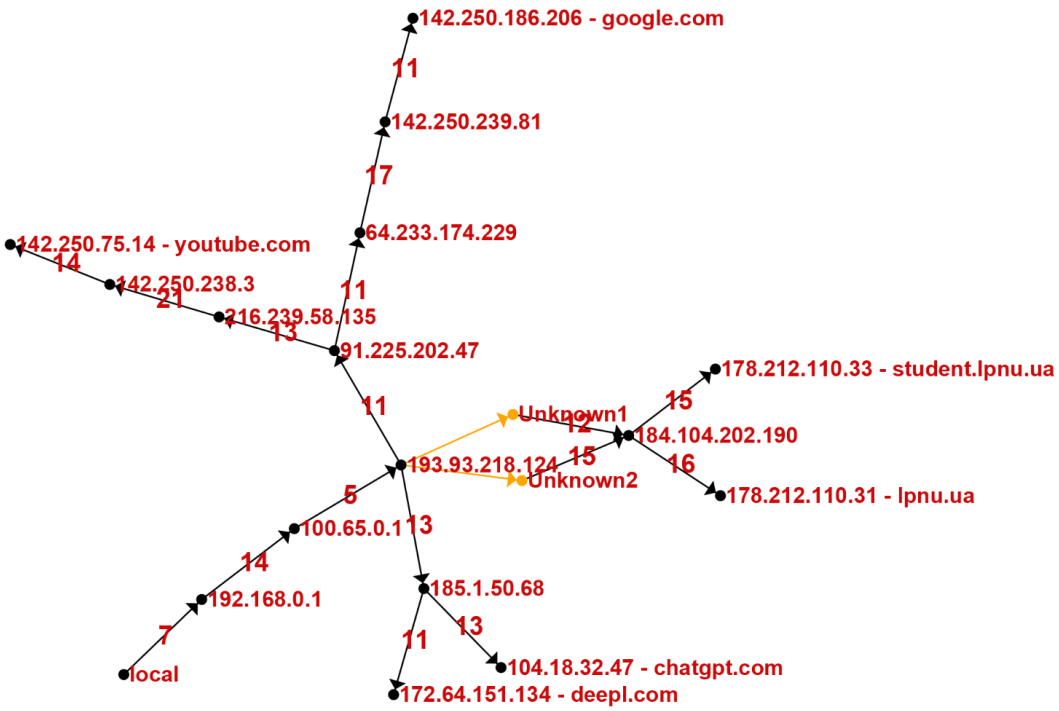


Fig. 4. Visualization of multiple traceroutes to six web resources: youtube.com, google.com, student.lpnu.ua, lpnu.ua, chatgpt.com, and deepl.com. RTT-oriented weighting model is used.

following these undefined ones (184.104.202.190) is shared. This suggests that Unknown1 and Unknown2 may physically correspond to the same router or a group of routers that block ICMP responses for security reasons.

Additionally, the graph demonstrates the logical cohesion of routes to google.com and youtube.com at node 91.225.202.47, which reflects their belonging to the Google infrastructure.

Analysis of the online tracing and visualization mode

The second experiment consisted of a series of studies aimed at evaluating the efficiency of the multithreaded online mode by performing simultaneous traceroutes to ten resources – google.com, lpnu.ua, dou.ua, youtube.com, rozetka.ua, chatgpt.com, wikipedia.org, deepl.com, facebook.com, and instagram.com – using different numbers of parallel threads. The main evaluation criteria were the graph construction speed (including the total traceroute execution time) and the quality of the obtained data, measured by the number of timeouts (**Table 1**).

Table 1. **Dependence of multithreaded mode efficiency on the number of parallel threads**

Number of threads	Graph construction time, s	Number of timeouts (unknown nodes)
1	65	2
3	26	3
5	16	4
7	15	5

The results demonstrate a clear trade-off between data collection speed and quality. As the number of threads increases from 1 to 7, the graph construction time decreases by more than fourfold (from 65 to 15 seconds), confirming the effectiveness of the parallel approach for real-time network monitoring.

The experiment with seven threads is particularly important: among five unknown nodes, four anomalous edges appear, which degrades the topological assessment quality. A detailed analysis of the resulting graph reveals a characteristic anomaly in the local network exit segment (**Fig. 5**).

In **Fig. 5**, a cluster of unknown nodes (Unknown1, Unknown2, Unknown3, Unknown4) can be clearly observed in close proximity to the local router 192.168.0.1. According to the principles of NAT routing, all devices within the local network should forward traffic through a single gateway 192.168.0.1, so the appearance of multiple "unknown" paths in this segment is an artifact.

An analysis of the textual traceroute logs reveals the exact mechanism behind these anomalies. In four out of ten traceroutes, timeouts were recorded at the first hop. The most probable cause is the activation of rate-limiting mechanisms on the local router 192.168.0.1 in response to the simultaneous arrival of seven intensive ICMP request streams.

A key advantage of the graph-based approach is its ability to visually identify anomalies quickly. In text format, determining that several unknown nodes physically correspond to the same router is extremely difficult and requires careful comparison of timeout positions across dozens of individual traceroutes. Graph visualization makes such patterns immediately apparent, as unknown vertices naturally cluster around the same transitions.

The experiments performed confirmed the high effectiveness of the proposed graph visualization approach to traceroute results for geolocation-independent network topology analysis. The system enables efficient detection of shared segments and critical nodes. The multithreaded online mode significantly expands the system's capabilities for real-time monitoring, providing a 3-4 times acceleration in data collection when properly configured.

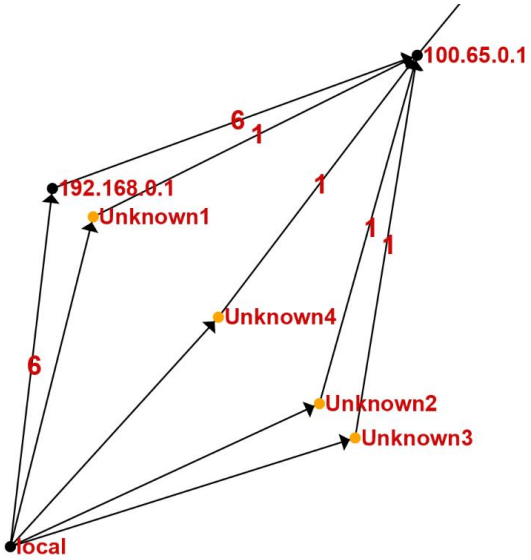


Fig. 5. Local network exit segment when using seven parallel tracing threads. Unknown nodes resulting from timeouts at the first hop are marked in yellow and labeled Unknown1–4.

Comparison with existing tools

To fully evaluate the capabilities of the proposed graph-based visualization approach, it is essential to compare it with existing traceroute analysis tools, both geographic and graph-oriented.

Geographic visualization systems such as Open Visual Traceroute [6] typically demonstrate higher rendering performance due to their reliance on external geolocation databases. Each IP address is directly mapped to predefined coordinates (latitude and longitude). As a result, the placement of nodes is determined by geographic metadata, and rendering complexity grows linearly with the number of nodes – O(n).

In contrast, the proposed method operates without geolocation data and constructs a layout solely from the textual output of the traceroute utility. It relies on a force-directed model – the Fruchterman-Reingold algorithm [11] – whose computational complexity is $O(n^2)$. This leads to lower rendering performance for graphs with a high number of nodes. However, geolocation independence provides an important benefit: the system remains fully functional even when IP-to-location mappings are unavailable or inaccurate. This makes the approach more versatile in research, diagnostic, and educational contexts where the primary interest lies in understanding routing behavior rather than geographic placement.

A direct comparison with VisTracer [10] is more challenging due to the absence of a publicly available implementation or reproducible benchmarks. The paper describing VisTracer indicates that its visualization pipeline partially relies on force-directed layout mechanisms but does not detail the algorithms or their computational cost. Conceptually, VisTracer is designed as a specialized analytical tool focused on detecting routing anomalies – particularly those associated with BGP hijacking, load balancing irregularities, and other forms of path instability. In contrast, the system presented in this work aims to provide a general-purpose graph-based representation of traceroute results.

Thus, the proposed approach complements VisTracer instead of competing with it, as they target distinct analytical objectives. While VisTracer investigates routing during data transmission, our approach focuses on analyzing the output of the traceroute utility. Consequently, our system is designed not to interfere with the traceroute acquisition process, but rather to effectively visualize its data.

Limitations of the proposed approach

The proposed method inherits several fundamental limitations from the traceroute utility itself. The accuracy of RTT measurements is affected by router-specific processing policies, ICMP rate-limiting mechanisms, and transient network conditions, such as increased evening load or temporary congestion on specific segments. These factors may distort the measured delays and reduce the reliability of RTT-based edge weighting. Additionally, ICMP packet filtering performed by firewalls or intermediate routers introduces missing hops and increases the number of unknown nodes. Both implemented strategies for handling such cases – explicit representation or skipping – may lead to incomplete structural representation or to the formation of edges that do not correspond to the actual physical routing path. Consequently, the resulting visualization should be interpreted with awareness of these limitations, particularly in networks with strict ICMP policies or highly dynamic routing behavior.

In addition, several limitations are associated with the visual representation of the resulting graphs. Large graphs may become visually overloaded, with overlapping node labels, edge length indicators, and occasional excessive edge crossings, all of which can hinder interpretability. While basic scaling and zooming mitigate these issues to some extent, they do not fully eliminate them. Further research could therefore focus on improving the layout and rendering algorithms to enhance readability and provide a more effective user experience.

CONCLUSION

Addressing the key tasks of monitoring routes to destination hosts and analyzing their timing characteristics, this work develops and experimentally validates a geolocation-independent approach to network analysis based on transforming the textual output of the traceroute utility into a unified weighted directed graph. A method for building the graph $G = (V, E)$ is proposed, enabling efficient aggregation of multiple traceroutes and accumulation of statistics for subsequent weighting. Two configurable policies for handling unknown nodes are implemented – explicit representation and skipping – along with two weighting models: an RTT-oriented model (for identifying segments with increased latency) and a frequency-based model (for highlighting shared or critical connections). A software module was developed with two operating modes: import of archived traceroute results and online data collection with multithreaded traceroute execution and incremental graph construction.

The obtained results demonstrate that combining multiple traceroutes into a single weighted graph reduces cognitive load compared to analyzing tabular logs, simplifies the identification of shared segments, bottlenecks, and atypical patterns (such as loops or diamonds). It has been shown that absolute RTT values, unlike per-hop increments, yield more stable edge weights in the presence of non-monotonic behavior caused by ICMP rate-limiting. In the multithreaded mode, a practical trade-off between speed and quality was established: using 3–5 threads provides significant acceleration of graph construction with only a moderate increase in timeouts, whereas more aggressive parallelism amplifies artifacts and reduces interpretability.

Unlike geographic systems that rely on external IP-to-location databases, the proposed method focuses on directly representing the routing paths derived from traceroute data and remains fully operational even when geolocation information is unavailable or unreliable.

Furthermore, the study shows that traceroute-based visualization is inherently constrained by the accuracy of RTT measurements and the presence of ICMP filtering, which may introduce missing hops and structural uncertainty in the resulting graph. As a result, the resulting graph structure should be interpreted with consideration of these factors, especially in networks with strict ICMP policies or highly dynamic routing behavior.

The scientific novelty of this work lies in combining the graph-based aggregation of multiple traceroutes with two alternative strategies for processing unknown nodes and implementing a graph weighting mechanism that provides an informative view of network behavior both in historical traceroute analysis and in real-time monitoring.

The practical value of the study is the creation of a network diagnostic and administration tool that enables quick detection of shared segments, critical nodes, and potential bottlenecks, thereby simplifying monitoring and troubleshooting. The method could also serve educational purposes, allowing students to interactively explore network routes and understand topology behavior through visual analysis.

Future work will focus on expanding the set of weighting metrics (e.g., response stability, packet loss frequency, etc.) and implementing automatic anomaly detection and highlighting mechanisms. Furthermore, a promising direction is the exploration of a hybrid RTT weighting scheme that combines both absolute and incremental delay values. Absolute RTT provides a stable representation of end-to-end latency, particularly in scenarios with non-monotonic behavior caused by rate limiting or router overload, while incremental RTT may reveal localized delays between specific hops. Integrating both perspectives could offer a more comprehensive view of latency distribution across the route and enhance the analytical value of the visualization.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, authorship, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [K.O., I.D.]; methodology, [I.D]; validation, [K.O., Z.S.]; formal analysis, [I.D.].; investigation [I.D.]; resources, [I.D]; data curation,[I.D.]; writing – original draft preparation, [I.D.]; writing – review and editing, [K.O., Z.S.]; visualization, [I.D.] supervision, [K.O.]; project administration, [Z.S.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Caiazza, C., Luconi, V., & Vecchio, A. (2019). TCP-based traceroute: An evaluation of different probing methods. Internet Technology Letters, 2(6), e134. https://doi.org/10.1002/itl2.134

[2] Luckie, M., Hyun, Y., & Huffaker, B. (2008). *Traceroute probe method and forward IP path inference.* In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC '08)* (pp. 311–317). Association for Computing Machinery. https://doi.org/10.1145/1452520.1452553

[3] Guo, H., & Heidemann, J. (2018, March). *Detecting ICMP rate limiting in the Internet.* In R. Beverly, G. Smaragdakis, & A. Feldmann (Eds.), *Proceedings of the 19th International Conference on Passive and Active Network Measurement (PAM 2018, Berlin, Germany)* (LNCS 10771, pp. 3-17). Springer. https://doi.org/10.1007/978-3-319-76481-8_1

[4] Li, W., Zhang, D., Xie, G., & Yang, J. (2005). TCP and ICMP in network measurement: An experimental evaluation. *Parallel and Distributed Processing and Applications, Lecture Notes in Computer Science*, 3758(3), 870–881. https://doi.org/10.1007/11576235_87

[5] Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., & Teixeira, R. (2006). Avoiding traceroute anomalies with Paris traceroute. *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06)*, 153–158. https://doi.org/10.1145/1177080.1177100

[6] VisualTraceroute. (n.d.). Visual traceroute. Retrieved from https://visualtraceroute.net/

[7] Sundin, S. (n.d.). Traceroute mapper. GitHub Pages. Retrieved from https://stefansundin.github.io/traceroute-mapper/

[8] Traceroute Online. (n.d.). *Traceroute online*. Retrieved from https://traceroute-online.com

[9] GeoTraceroute. (n.d.). GeoTraceroute. Retrieved from https://geotraceroute.com/

[10] Fischer, F., Fuchs, J., Vervier, P.-A., Mansmann, F., & Thonnard, O. (2012). VisTracer: A visual analytics tool to investigate routing anomalies in traceroutes. *Proceedings of the Ninth International Symposium on Visualization for Cyber Security (VizSec '12)*, 80–87. https://doi.org/10.1145/2379690.2379701

[11] Cheong, S.-H., & Si, Y.-W. (2022). *Force-directed algorithms for schematic drawings and placement: A survey. arXiv preprint* arXiv:2204.01006. https://arxiv.org/abs/2204.01006

[12] Obelovska, K., Danych, I. (2022). Adoption of the OMNET++ Simulator for the Computer Networks Learning: A Case Study in CSMA Schemes. In: Hu, Z., Zhang, Q., Petoukhov, S., He, M. (eds) Advances in Artificial Systems for Logistics Engineering. ICAILE 2022. Lecture Notes on Data Engineering and Communications Technologies, vol 135. Springer, Cham. https://doi.org/10.1007/978-3-031-04809-8_21
[13] Microsoft. (n.d.). tracert. In *Windows Server: Windows commands*. Microsoft Learn. https://learn.microsoft.com/uk-ua/windows-server/administration/windows-commands/tracert
[14] Postel, J. (1981). Internet control message protocol. RFC 792. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc792
[15] Baker, F. (1995). Requirements for IP version 4 routers. RFC 1812. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc1812
[16] Danych, I. M., & Shpak, Z. Y. (2025). Methods and tools for graph visualization in dynamic systems: An experimental study. Ukrainian Journal of Information Technology, 7(1), 131–139. https://doi.org/10.23939/ujit2025.01.131

ВІЗУАЛІЗАЦІЯ НА ГРАФАХ РЕЗУЛЬТАТІВ УТИЛІТИ TRACEROUTE ДЛЯ АНАЛІЗУ КОМП'ЮТЕРНИХ МЕРЕЖ

Іван Данич, **Квітослава Обельовська**, **Зореслава Шпак***
Національний університет «Львівська політехніка»,
вул. Степана Бандери 12, м. Львів, 79013 Україна

АНОТАЦІЯ

Вступ. Моніторинг досяжності вузлів, виявлення маршрутів і аналіз затримок є важливими для забезпечення продуктивності, надійності та безпеки мережі. Утиліта traceroute широко застосовується для діагностики шляхів, однак її текстовий формат ускладнює аналіз складних топологій. Фільтрація ICMP-трафіку, обмеження швидкості відповідей і балансування навантаження призводять до пропусків хопів, немонотонних затримок і артефактів, що важко виявити у табличному вигляді. Існуючі методи візуалізації базуються на геолокаційних даних і зосереджуються на географічних аспектах, а не на топології мережі. Тому актуальним є створення геолокаційно незалежного інструменту візуалізації результатів утиліти traceroute у вигляді графа.

Матеріали та методи. Запропоновано підхід, що перетворює одиничні та множинні результати утиліти traceroute у зважений орієнтований граф, де вершини є IP-вузлами. Невідомі вузли обробляються двома способами: явним маркуванням або пропуском із прямим з'єднанням відомих вузлів. Ребра зважуються за частотою появи (для виявлення критичних зв'язків) і середнім RTT (для визначення ділянок із затримками). Візуалізацію виконано за алгоритмом Fruchterman-Reingold. Система реалізована на Java з використанням JavaFX, Spring Core і GraphStream, підтримує імпорт архівних даних і трасування у режимі реального часу через багатопотокову модель producer-consumer, що забезпечує цілісність графа.

Результати. Об'єднання трасувань до кількох цілей в один граф дає змогу швидко знайти спільні сегменти та критичні вузли. Виявлено немонотонність RTT, ймовірно спричинену rate limiting, що обґрунтовує використання абсолютних, а не приростових значень RTT. У проведених експериментах багатопотоковий режим прискорив побудову графа у чотири рази при семи потоках, але збільшив кількість таймаутів. Використання п'яти потоків забезпечило оптимальний баланс швидкості та якості.

Висновки. Запропонований підхід підвищує ефективність аналізу мережі, формуючи геолокаційно незалежний граф для діагностики. Розроблена система зменшує когнітивне навантаження порівняно з аналізом табличних даних і пришвидшує виявлення потенційних «вузьких місць», спільних сегментів, критичних вузлів та аномалій маршрутизації.

Ключові слова: traceroute, візуалізація на графах, мережева топологія, мережевий моніторинг, зважені орієнтовані графи.

Received / Одержано 06 November, 2025 | Revised / Доопрацьовано 01 December, 2025 | Accepted / Прийнято 02 December, 2025 | Published / Опубліковано 25 December, 2025

Electronics and Information Technologies, 2025, 32, 19–38
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.032.26

INFORMATION TECHNOLOGY OF AUTONOMOUS NEUROFUZZY MOTION CONTROL OF THE GROUND MOBILE ROBOTICS PLATFORM

***Ivan Tsmots*[1]*, *Vasyl Teslyuk*[1], *Yurii Opotyak*[1], *Vasyl Rabyk*[2], *Oleksandr Oliinyk*[1]**
[1]Lviv Polytechnic National University,
12 Stepan Bandera St., Lviv 79013, Ukraine
[2]Ivan Franko National University of Lviv,
1 Universytetska St, Lviv, 79000, Ukraine

Tsmots, I., Teslyuk, V., Opotyak, Yu., Rabyk, V. & Oliinyk, O. (2025). Information Technology of Autonomous Neurofuzzy Motion Control of the Ground Mobile Robotics Platform. *Electronics and Information Technologies*, 32, 19–38. https://doi.org/10.30970/eli.32.2

ABSTRACT

Background. The development of ground-based mobile robotic platform (GBMRP) motion control systems requires ensuring autonomy, adaptability to dynamic environmental changes, and consideration of limitations on computing resources, mass, dimensional parameters, and energy consumption. Traditional methods of GBMRP motion control, based on accurate mathematical models, are ineffective in real dynamic conditions, where it is impossible to accurately describe the state of the GBMRP and its environment. Therefore, for GBMRP motion control in such situations, the development of an appropriate information technology based on fuzzy logic and neural networks is proposed.

Materials and Methods. Information technology for autonomous neurofuzzy control of the GBMRP movement is developed based on integrated, hybrid, and problem-oriented approaches using component-oriented technology. Fuzzy logic methods will ensure the flexibility and stability of the control system in the presence of noise, measurement errors, and unpredictable environmental changes, and enable integration with neural networks to increase accuracy and speed. Neural network methods offer enhanced accuracy in navigation measurements, data recovery, and the construction of a neural-like defuzzifier, resulting in improved control signal formation.

Results and Discussion. An information technology for autonomous neurofuzzy control of the GBMRP movement has been developed, which, through the use of methods and means of collecting, storing and processing navigation data under conditions of interference and incomplete information, a combination of fuzzy logic and neuro-like structures based on the Sequential Geometric Transformations Model, has ensured adaptability and decision-making under conditions of uncertainty, and the accuracy of the formation of control signals. A fuzzy logic rule base has been developed, providing an adequate representation of expert knowledge in the control system.

Conclusion. In this work, an information technology for autonomous neurofuzzy control of the GBMRP in dynamically changing situations has been developed, based on integrated, hybrid, and problem-oriented approaches using component-oriented technology, while considering limitations on computing resources and energy consumption.

Keywords: information technology, fuzzy logic, neural network, ground mobile robotics platform, platform motion control, defuzzification

© 2025 Ivan Tsmots et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

The current stage of robotics development is characterized by an increase in requirements for autonomy, adaptability, and intellectuality of information technology tools for controlling ground-based mobile robotic platforms (GBMRP). Such means should provide autonomous control of the movement of the GBMRP under conditions of uncertainty, minimal predictability of the environment, the impact of external disturbances, and limitations on computing resources, mass-dimensional parameters, energy consumption, and cost. Therefore, a pressing issue is the development of information technology for autonomous traffic control of the GBMRP, which will meet the listed requirements.

To date, numerous studies and publications have been conducted on the development of information technology components for autonomous neurofuzzy movement control of GBMRP [1,2]. The primary objective of these studies is to develop effective components for the information technology of autonomous neurofuzzy motion control of the GBMRP, aiming to synthesize a real-time movement control system for the GBMRP that meets constraints on dimensions, power consumption, and cost.

The analysis of the work [3,4] showed that the uncertainty of the external environment in which the GBMRPs operate necessitates the inclusion of a set of intelligent sensors and neural network data processing tools in their composition, which should enable autonomous and safe control of the platform's movement.

One of the most important areas of research is the development of neural network algorithms for the GBMRP motion control system [5,6]. In these works, the authors propose a system of neural network control of the movement of the GBMRP, which provides movement in conditions of uncertainty and minimal predictability of the environment. The articles [7,8] consider the use of deep learning methods for developing neural network control systems for the GBMRP.

The development of mobile platform control systems is the combination of fuzzy logic with the adaptive capabilities of artificial neural networks [9,10]. The paper [11] shows that Takagi–Sugeno controllers with neural adaptation can effectively implement real-time navigation and obstacle avoidance strategies. Such control systems combine the ability to learn from experience with logical interpretation of rules, which makes them suitable for mobile platforms with uncertain or partially known environmental models.

In further research, adaptive neuro-fuzzy inference systems (ANFIS) were proposed, which have become the standard in tasks involving local navigation and stabilization of robot movement. In particular, [12, 13] demonstrated the effectiveness of ANFIS controllers for reactive navigation and obstacle avoidance based on sensor data (laser, ultrasound, GPS). Such approaches ensure smooth trajectories and noise resistance of sensors through a combination of fuzzification, fuzzy rules (such as TSK, proposed by Takagi, Sugeno, and Kang), and training of membership function parameters.

The current direction in the development of neurofuzzy systems is to create hybrid architectures that combine fuzzy logic with deep learning and reinforcement learning methods. Papers [14, 15] demonstrate the feasibility of learning fuzzy rules in environments with variable dynamics using deep reinforcement learning (DRL+ANFIS). This approach enables you to optimize not only instantaneous reactions but also long-term performance criteria, including trajectory time, power consumption, and resistance to dynamic interference.

Of particular interest is the SGTM, which is being developed in Ukrainian scientific schools and considers the process of information processing in neurolike structures as a sequence of geometric transformations in the feature space [16, 17]. This approach is related to modern ideas in geometric deep learning, as it enables the formation of transformation-invariant data representations. The use of SGTM in the motion control systems of a mobile robotic platform enables the development of non-iterative learning algorithms and achieves zero methodological error during defuzzification, which is particularly important for real-time implementation on hardware platforms (FPGA, SoC, SBC) [18, 19].

The practical implementation of neurofuzzy controllers necessitates consideration of the limitations imposed by real-time and hardware resources. As noted in [20], for mobile robots based on microcontrollers or single-board computers, a balance between accuracy, speed, and energy efficiency is important. Hardware implementations on FPGAs enable the implementation of cascades of fuzzy rules in pipeline mode, resulting in minimal delays in processing touch data [21, 22]. The articles [23, 24] demonstrate that the use of the tabular-algorithmic method for implementing basic operations in neurofuzzy motion control of a mobile platform results in improved performance and reduced computing resource requirements.

Therefore, an urgent task is to develop the components of the information technology for autonomous neurofuzzy movement control of the GBMRP, which is based on fuzzy logic and SGTM-based neural networks, and to synthesize systems of neurofuzzy control for the movement of the GBMRP on this basis.

MATERIALS AND METHODS

Traditional methods of classical control of the movement of the GBMRP, which are based on accurate mathematical models, are ineffective in real dynamic conditions, where it is impossible to accurately describe the state of the GBMRP and the environment. To control the movement of the GBMRP in such conditions, on the basis of fuzzy logic and neural network methods, a control system is created, which must be able to build a route and control the parameters of movement (set the speed of movement and angles of rotation).

Fuzzy logic provides formalization of expert knowledge in the form of linguistic if-then rules and decision-making based on incomplete or inaccurate sensory data. The use of fuzzy logic in the GBMRP control system will ensure the flexibility and resilience of the control system to noise, measurement errors and unpredictable changes in the environment; provide effective management without the need to create an accurate mathematical model of the object and environment; high adaptability to dynamic changes and uncertainty; smooth adjustment and absence of sharp fluctuations in control signals; ease of implementation of the rule base and the possibility of its further configuration or training; the ability to integrate with neural networks to improve the accuracy and speed of the system.

One of the ways to achieve high technical and operational characteristics of the means of information technology for controlling the movement of GBMRP is the use of neural networks based on the paradigm of the Successive Geometric Transformations Model (SGTM) for evaluating data from sensors in conditions of interference and incomplete information, building a neural defuzzifier with increased accuracy of generating signals for controlling the drive of the GBMRP. A feature of SGTM-based neural networks is the fundamental possibility of non-iterative calculation of the weights of synaptic connections between neural elements, as well as the use of a tabular algorithmic method for their implementation.

It is advisable to develop an information technology for autonomous neurofuzzy control of the movement of GBMRP based on integrated, hybrid, and problem-oriented approaches using component-oriented technology. Therefore, the development of information technology for autonomous neurofuzzy motion control of the GBMRP is a pressing scientific and practical task aimed at enhancing the level of intelligence, reliability, and technical and operational characteristics of the GBMRP movement control system.

The object of research is the processes of collecting, storing, and processing navigational data, as well as the neurofuzzy control of the movement of GBMRP. The subject of the study is the models, methods, algorithms, and structures of information technology tools for autonomous neurofuzzy control of the movement of the GBMRP in real-time.

The purpose of this work is to develop an information technology for autonomous neurofuzzy control of the movement of GBMRP, which operates under conditions of uncertainty, minimal predictability of the environment, and the influence of external disturbances, while also considering limitations on computing resources, mass-dimensional parameters, and energy consumption.

To achieve this goal, the following main objectives of the study are defined:

- development of the structure of information technology for autonomous neurofuzzy control of the movement of the GBMRP;
- means of collecting and storing navigational data on the state of the GBMRP environment;
- methods and tools for processing navigation data;
- methods and means of route planning for GBMRP;
- models and means of collision avoidance;
- methods and means of fuzzification of navigation data;
- a base of rules of fuzzy logic for controlling the movement of GBMRP;
- methods and means of forming fuzzy conclusions;
- methods and means of neuroid defuzzification;
- tabular and algorithmic method of increasing the speed of the means of neurofuzzy control of the GBMRP;
- kinematic model of GBMRP.

RESULTS AND DISCUSSION

Development of the information technology structure for autonomous neurofuzzy movement control of GBMRP

The development of information technology for autonomous neurofuzzy control of the movement of GBMRP in conditions with a dynamic change in the situation is proposed to be carried out based on an integrated approach, which includes:

- methods and means of collecting navigational data on the dynamic state of the GBMRP environment;
- methods and means of storing navigation data on the dynamic state of the environment of the GBMRP;
- methods and means of processing and evaluating navigation data from sensors in conditions of interference and incompleteness of information;
- methods and means of planning the route of the GBMRP;
- methods and means of fuzzy control of the movement of the GBMRP;
- bases of rules for unclear control of the movement of the GBMRP;
- methods and means of training and functioning of the neural network in the process of defuzzification;
- tabular and algorithmic method of increasing the speed of the GBMRP motion control means;
- modern element base (microcontrollers, systems on a chip, programmable logic integrated circuits such as FPGAs, etc.);
- means of computer-aided design of software and hardware.

The development of information technology tools for autonomous neurofuzzy control of the movement of GBMRP is proposed to be carried out using component-oriented technologies. The essence of component-oriented development technology is to divide the means into independent, interconnected functional components (modules), which can be developed, tested, and improved separately, and then integrated into a single system through standardized interfaces.

An approach based on this technology provides modularity and scalability of the architecture, reuse of software and hardware solutions, reduction of development time and implementation costs, and the ability to flexibly adapt the system to new conditions or update the technological base.

The component-oriented design technology is based on the following principles: functionality encapsulation, where each component implements its own logic independently of the

others; use of standardized interfaces; component interchangeability, where updating or replacing a separate module does not require a complete redesign of the system; hierarchy of architecture, in which components can be combined into subsystems (navigational, analytical, executive, etc.); support for reuse, in which finished components can be integrated into other robotics platforms; hardware-software coordination, which ensures correspondence between software components and physical modules of the equipment.

According to component-oriented development technology, the GBMRP motion control system comprises several functional components, each of which performs well-defined tasks and has a standardized interface for interaction with other subsystems. For the synthesis of the GBMRP motion control system, the following components can be used: sensory data collection, neural network accuracy improvement, data processing, and fuzzy logical decision-making.

The structure of the information technology for neurofuzzy control of the GBMRP movement in dynamic situations is shown on **Fig. 1**.

To control the movement of GBMRP, we will use a hybrid approach that integrates: fuzzy logic for decision-making in conditions of uncertainty and the formation of adaptive control actions; neuro-like structures for the implementation of fast computing; and hardware and software for adaptive motion control in real time.

The use of fuzzy logic will ensure decision-making in the face of fuzzy or contradictory data, as well as the smoothness and continuity of control actions. With neurofuzzy control of the GBMRP, along with fuzzy logic, neuro-like structures utilizing the SGTM paradigm are employed, which provide speed, repeatability, and the ability to operate with large volumes of data. The SGTM paradigm is based on a non-iterative approach to training neuro-like structures, which involves the direct calculation of weights during a gradual decrease in the dimensionality of the space of input multidimensional data on neurons of the hidden layer.

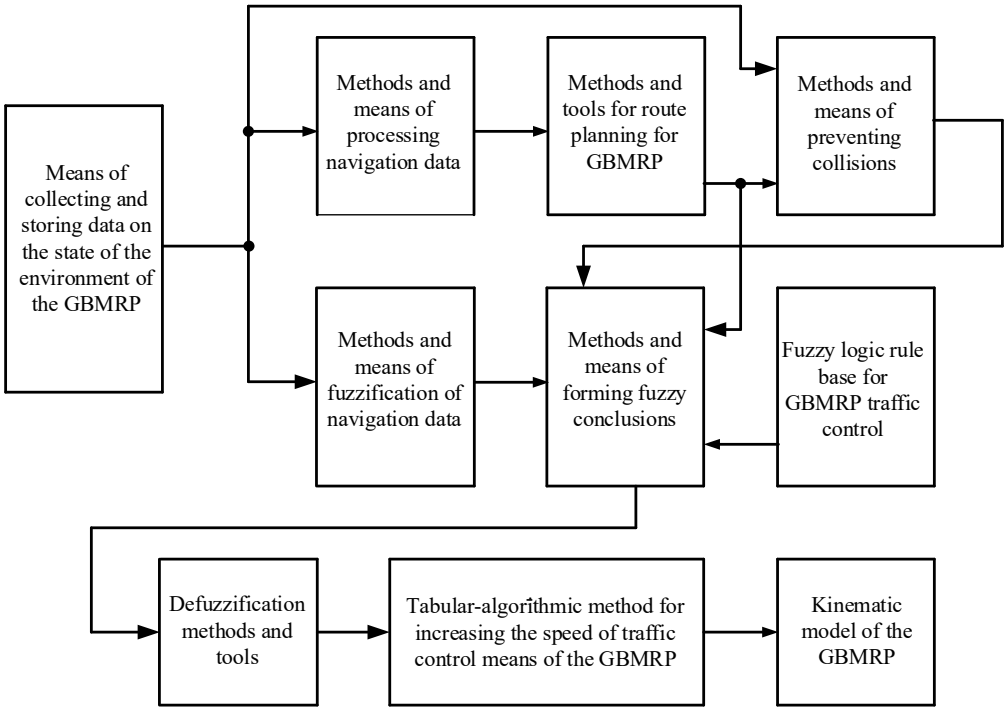


Fig. 1. Structure of information technology for autonomous neurofuzzy motion control of GBMRP.

The implementation of information technology for neurofuzzy control of the movement of GBMRP is proposed to be carried out based on a problem-oriented approach, which involves a combination of software and specialized hardware. The process of interpenetration of software (universal) and hardware (specialized) ensures their implementation with high technical and operational characteristics.

The development of means of information technology for controlling the movement of the GBMRP is proposed to be carried out according to the following principles: variability of the equipment composition, which provides for the presence of a processor core and replaceable specialized means, with the help of which the core is adapted to the requirements of a specific application; modularity; pipeline and spatial parallelism of data processing; openness of software; specialization and adaptation of hardware to the structure of data processing algorithms.

The main stages of the implementation of the information technology of neurofuzzy control of the movement of GBMRP are:

- means of collecting and storing data on the state of the environment of the GBMRP;
- methods and tools for processing navigation data;
- methods and means of route planning for GBMRP;
- methods and means of fuzzification of navigation data;
- a base of rules of fuzzy logic for controlling the movement of GBMRP;
- methods and means of forming fuzzy conclusions;
- methods and means of defuzzification;
- tabular and algorithmic method of increasing the speed of the means of neurofuzzy control of the GBMRP;
- kinematic model of GBMRP.

Means of collecting and storing navigational data on the state of the environment of the GBMRP.

With neurofuzzy control of GBMRP movement, it is necessary to continuously monitor the navigation state of the environment, measure distances to obstacles, determine coordinates, and control the direction and speed of the GBMRP movement.

To collect and store navigation data on the state of the GBMRP environment, the following tools may be used: Raspberry Pi microcomputer, ESP32-C3 microcontroller, MPU-6050 gyroscope, QMC5883L digital compass, YDLidar X4 lidar, GPS module, and a set of sensors for collecting climatic parameters. Each sensor must have dedicated computing resources for processing the received data. Their implementation is possible using modern microcontrollers that have the appropriate sets of interfaces (SPI, I^2C, Serial) and are equipped with sufficient computing resources to ensure interaction with the sensors and implement the necessary data pre-processing in real-time. The collection and storage of navigation data on the state of the environment of the GBMRP is controlled by the Raspberry Pi microcomputer, the information from which is transmitted to the control system.

The navigation data used to control the movement of the GBMRP is stored in a database implemented on a single-board Raspberry Pi computer. The type of database affects both the performance and scalability of the hardware and software solution, as well as its ability to efficiently analyze, process, and manage data. Traditional relational databases, such as PostgreSQL, MySQL, or SQL Server, are robust and have extensive functionality, but they require more system resources and are not optimized for operations that are common in IoT and GBMRP. These databases are optimized for processing complex queries and providing transactional sequences, but they are not efficient when working with large amounts of data received from navigation sensors that arrive at high intensity. The advantages, for example, of the InfluxDB database are its ability to efficiently process large amounts of timestamped data. Working with data in the form of time series

is used to prepare datasets for machine learning models when predicting future states based on historical data.

Methods and tools for processing navigation data.

Methods and tools for processing navigation data in GBMRP are a key component of ensuring the autonomy, accuracy, and safety of mobile platform movement in complex environments. The main methods of processing navigation data in the GBMRP are:

- localization methods that are designed to determine the position and orientation of the GBMRP in space;
- route planning methods that are designed to find the optimal path of the GBMRP, taking into account the map and obstacles;
- methods of avoiding obstacles in real time, which are designed for dynamic adaptation of the movement of the GBMRP to new circumstances;
- sensor integration techniques that combine data from different sensors to improve accuracy and reliability;
- artificial intelligence techniques that enable environmental perception, decision-making, route planning, and adaptation to changes in the environment.
- methods and means of processing navigation data in the GBMRP with autonomous navigation, which operate in conditions with dynamic change of situations, interference, and incompleteness of information, should provide the following requirements:
  - autonomous navigation of the GBMRP, which will provide independent route planning, obstacle avoidance, and recovery from failures;
  - high accuracy of positioning of the GBMRP;
  - reliability of navigation data, i.e., the ability to work in conditions of noise, signal loss, and temporary sensor failures;
  - integration of navigation data from various sources;
  - adaptability to changes in the environment of the GBMRP;
  - processing navigation data and real-time management decision-making;
  - building environmental maps;
  - orientation of intelligent algorithms to built-in problem-oriented systems with limited computing resources;
  - reduction of energy consumption, dimensions, and weight.

Neural-like means of recovering lost navigation data. To recover the lost navigation data, we will use a neural network based on the SGTM paradigm. Since the data for recovery from a typical sensor is overwhelmingly presented in the form of a time series, we can use the time window method to process it. In this case, the number of neural network inputs equals the selected time window size.

As a result of neural network processing, forecasting (reproducing lost) data for the next countdown is performed. The quality of reproduction in this case depends on the number of neurons in the hidden layer; however, a compromise between the accuracy of training and the accuracy of prediction (reproduction of lost data) should be considered in this context. The optimal values of this parameter are determined by the dataset data, their number, and structure.

A significant advantage of SGTM-based neural networks over others is their ability to learn quickly and non-iteratively. Additionally, these neural networks effectively replicate the lost data for the reference group.

Means of intelligent processing of navigation data. Means of intelligent processing of navigation data should provide an increase in the accuracy of measuring motion parameters, predicting geographical coordinates, and facilitating effective interaction with the environment despite interference.

The means of intelligent processing of navigation data in the GBMRP consist of the following modules: wireless communication and data protection; processing, analysis and

recovery of lost data; neural network improvement of the accuracy of measuring movement parameters; neural network improvement of the accuracy of determining geographic coordinates; neural network forecasting of geographical coordinates and route of movement of the GBMRP; data collection and storage; determination of geographical coordinates and control of the movement of the GBMRP.

Methods and means of intelligently improving the accuracy of the measurement of navigation parameters. The task of increasing the accuracy of data obtained from various navigation sensors in the conditions of interference and incomplete information is important for the GBMRP. The complexity of solving such problems lies in the fact that the accuracy at the output of the sensors is a consequence of the influence of many factors, each of which has a different degree of influence on the result.

The task is especially challenging when it is impossible to clearly distinguish the factors influencing the system and when the magnitude of their influence on the resulting parameter is unknown. Navigation sensors, which are widely used in many systems, are particularly indispensable for GBMRP.

The measurement accuracy of such sensors is affected by noises that are caused by both their design and the external environment. It is proposed to utilize neural network singular spectral analysis (SSA) to enhance the accuracy of navigation sensor measurements, which enables the detection and removal of noise from the output signal. Using the neural network SSA, an information technology has been developed to improve the accuracy of measuring the parameters of movement and spatial orientation of the GBMRP. The main stages of such information technology are:

- collection and storage of data from navigation sensors;
- pre-processing of navigation data;
- determination of neural network parameters based on SGTM for singular spectral analysis;
- development of a neural network based on SGTM for SSA;
- detection and removal of noise in the measuring output signal;
- development of a neural network structure based on SGTM for measuring the navigation parameter with increased accuracy;
- setting up a neural network based on SGTM and calculating weighting factors;
- calculation of tables of macropartial products for tabular and algorithmic implementation of neuroelements;
- problem-oriented implementation of a neural network based on SGTM to measure the navigation parameter with increased accuracy.

For the implementation of neural network SSA, neural networks based on the SGTM paradigm with projective and ordered lateral connections were selected, which are fast, non-iterative, do not accumulate errors, and do not have limitations on measurability. The use of such neural networks for implementing neural network SSA will ensure real-time operation and achieve high technical and operational characteristics.

Forecasting the movement of GBMRP. There are various algorithms for predicting the route and movement of the GBMRP. The choice of a specific algorithm is carried out taking into account various conditions. In particular, the GBMRP operates under deterministic or non-deterministic, static or dynamic conditions.

If the GBMRP operates in static conditions, then the forecasting of motion is reduced to finding a sequence of points in the trajectory of the possible movement of the GBMRP, with the possibility of conflict-free obstacle avoidance. If dynamic changes in the external environment are possible during the movement of the GBMRP, then, in addition to finding trajectory points, it is necessary to regulate the speed of movement on specific sections of the route.

Algorithms for global planning of the movement of GBMRP in deterministic conditions contain information about the external environment. Therefore, it is possible to identify

specific areas where the movement of the GBMRP is feasible and then select the optimal path. A problem in predicting the movement of the GBMRP can be solved using both accurate and heuristic algorithms.

The main disadvantage of accurate algorithms is the high computational complexity, and heuristic algorithms do not guarantee the completeness of the search for the optimal path. The advantages of heuristic algorithms include reducing computational complexity and sensitivity to data errors. Some of these algorithms focus on finding the optimal path. At the same time, they read the data from the map of the external environment and consistently try to find the best way for the movement of the GBMRP, avoiding obstacles. Such algorithms are effective even when there are a large number of various obstacles in the way of GBMRP.

Despite this, in real-world conditions, the GBMRP operates in uncertain environments, and the solution to the motion forecasting problem cannot be carried out by either accurate or heuristic algorithms. Therefore, in this case, the best option is to use artificial intelligence tools. Currently, the most common methods for predicting the movement of GBMRP are neural network algorithms.

In general, the following requirements are set for neural network algorithms for predicting the movement of GBMRP:

- minimizing the passage time;
- minimizing the length of the path;
- minimization of deviations from a given trajectory;
- reliability (absence of algorithm failures);
- ensuring the impossibility of collisions with obstacles and other GBMRPs.

Under conditions of uncertainty, the movement of the GBMRP can be predicted with the help of artificial neural networks. When implementing motion forecasting algorithms, it is essential to consider the dimensions of the GBMRP.

The advantage of using a neural network approach for implementing GBMRP motion prediction is the possibility of parallel signal processing. This is achieved by combining a large number of neurons into layers and connecting neurons across different layers in a specific manner. Also, artificial neural networks provide the ability to learn.

The solution of the problem of GBMRP forecasting based on the neural network approach is reduced to the following stages:

- formalization of the planning task;
- network topology selection;
- displaying the interaction of network neurons in the form of a neural map (surface);
- calculation of the full trajectory in the form of some procedure of "ascent" to the top of the surface (target).

The conditions for applying this approach are determined by the formalization of the motion forecasting problem. The input data consists of environmental data collected from sensors hosted on the GBMRP. Based on the information obtained, the configuration of the given space and the location of obstacles are determined. Determining the exact configuration of the workspace largely depends on the technical capabilities of the sensor system, so it is necessary to use high-quality sensors to obtain the necessary information.

The main advantages of using neural network algorithms for predicting the movement of GBMRP include the possibility of their hardware implementation in the form of neuro-accelerators. Since neural networks can quickly adapt to changes, they can be used to predict the movement of GBMRP in dynamic, non-stationary environments.

Methods and tools for route planning for GBMRP

Planning of GBMRP routes is a multifactorial task that should take into account groups of the following factors:

- environment – static, dynamic obstacles and road conditions;
- technical limitations of the GBMRP – battery reserve, maximum carrying capacity, movement speed, ability to overcome obstacles, type of movement;
- sensors and navigation systems – lidar, video camera, angular velocity meter, digital compass, and GPS;
- route restrictions – minimizing distance, avoiding dangerous areas, minimizing the time to complete the task, and taking into account priorities.

Geometric methods can be used to plan the route of the GBMRP, which are grouped on a mathematical model of the environment using geometric calculations to find the optimal route. Such methods include the graph method of GBMRP route planning, which involves using graphs to model space and finding the optimal path between given points. Building a navigation graph for planning the route of the GBMRP involves the following stages:

- modeling space as a set of nodes and edges, where nodes are points at which the GBMRP can stop or change direction, and edges are the paths along which the GBMRP can move;
- assigning each edge a weight that characterizes the costs necessary for its passage (length, time, energy consumption, etc.);
- optimization of the graph by eliminating unnecessary nodes and clarifying weights, taking into account dynamic conditions (moving obstacles).

To plan the route of the GBMRP in complex environments, we will utilize the Navigation Stack package, which is specifically designed to integrate with navigation systems within the ROS 2 (Robot Operating System) framework. The main components of the Navigation Stack package are:

- Global Planner, which provides a path from the start to the end point based on a global map using the A* or Dijkstra algorithm;
- Local Planner, which provides real-time obstacle avoidance and trajectory correction based on sensor data using the DWA (Dynamic Window Approach) or TEB (Timed Elastic Band) algorithm.

Planning the route of the GBMRP using the Navigation Stack package involves setting the coordinates of the start and end point (target) for the ride. Global Planner used a map for optimal path planning to the target. The Local Planner takes into account dynamic obstacles and adjusts the route in real-time.

Collision Avoidance Models and Tools.

An important element of the GBMRP traffic control system is the collision avoidance subsystem, which utilizes safety zones and Petri nets. The safety zone for GBMRP defines the area around the platform where collisions are avoided, ensuring a safe distance between the platform and other objects in the environment.

The main factors that affect the size of the safety zone are: the size of the GBMRP itself and the cargo it carries; the speed of movement of the GBMRP; the response time of the sensors, which is required to detect an obstacle; the braking distance, which depends on the mass of the GBMRP with a load and the power of the brake system. The diameter of the safety zone D for the GBMRP is calculated according to the formula:

$$D = vt_r + \frac{v^2}{2a} + M, \tag{1}$$

where v is the speed of movement of the GBMRP (m/s), t_r is the sum of the reaction time of sensors and data processing (s), a is the acceleration of braking (m/s²), and M is the additional margin.

Simulation models using safety zones to avoid collisions during the movement of GBMRPs are based on the principle of creating virtual constraints that define a safe space around the platform. By utilizing safety zones, GBMRPs can make informed decisions about their routes and actions in real-time, thereby avoiding collisions with both static and dynamic obstacles.

To prevent collisions of GBMRP with obstacles, it is proposed to use a model based on the theory of Petri nets. In mathematical form, the collision avoidance model based on the theory of simple Petri nets can be written as follows:

$$Mod\_CA =< P, T, M_0 >, \tag{2}$$

where Mod_CA is the collision avoidance model based on the theory of simple Petri nets, P is the set of positions, T is the set of transitions, and M_0 is the initial marking.

The development of such a model involves the use of priorities and safety zones defined by the GBMRP, the size of which is determined by formula (1). In such a model, when a dynamic obstacle (such as vehicles and people) violates the safety zone of the GBMRP, a priority mechanism is proposed to resolve the conflict situation. When moving the GBMRP to ensure effective coordination of their movement, avoiding collisions, and achieving a common goal, the urgent task is to assign priorities to vehicles, people, and the GBMRP.

The following parameters can be used to determine the priority of the GBMRP: distances to obstacles, type of obstacle (static or dynamic), features of the dynamic obstacle (such as vehicles or people), and occupancy of traffic sectors. When controlling traffic on the GBMRP using the priority mechanism and safety zone, it is assumed that vehicles and people within the safety zone have a higher priority than those on the GBMRP.

Therefore, the GBMRP must slow down and give way to vehicles and people who have higher priority. In the event of the signal indicating a violation of the security zone disappearing, the platform continues to perform the previously assigned task.

A collision avoidance model considering the priority mechanism and safety zone based on colored Petri nets is written as follows:

$$Mod\_col\_CA =< P, T, F_{in}, F_{out}, M_0, TypeP, TypeM >, \tag{3}$$

where Mod_col_CA is a collision avoidance model taking into account the priority mechanism and the safety zone based on colored Petri nets, P is a set of positions, T is a set of transitions, F_{in} is a set of input arcs, F_{out} is a set of output arcs, M_0 is the initial marking, $TypeP$ is the types (colors) of markers that may be present in a particular position, $TypeM$ is the types of markers.

In the model, we have two types of markers $TypeM\_A$ and $TypeM\_B$. The presence of a $TypeM\_B$ marker in the position signals that the GBMRP resolves the conflict situation.

Methods and means of fuzzification of navigation data

Fuzzification of data from navigation sensors (GPS, gyroscopes, accelerometers, digital compasses, LiDAR) involves converting this data into fuzzy sets or linguistic variables that are understandable to humans. The peculiarity of such a transformation is the formation of belonging functions and the determination of the number of terms of the linguistic variable.

The choice of the membership function depends on the following characteristics of the navigation data: noise, data dynamics, and value distribution. When developing a motion control system for the GBMRP, the following belonging functions are most often used: triangular, trapezoidal, Gaussian, sigmoidal, and parabolic. For real-time navigation data

fuzzification, it is advisable to choose a triangular function that is easy to implement. The main methods used for the fuzzification of navigation data are: the method of equal degrees, discrete fuzzification, continuous fuzzification, the method of splitting by parameters, and the combined method. Of the listed methods for fuzzification in fuzzy motion control systems of the GBMRP, it is most expedient to use the method of continuous fuzzification and the combined method. The continuous fuzzification method provides a smooth transition between different levels of fuzzy affiliations. This method works with fuzzy sets, which allow elements to belong to a set with a certain degree of belonging (values from 0 to 1). Continuous fuzzification reduces the impact of sudden changes in the system's inputs, making it more resistant to noise.

After the fuzzification is completed, specific values of belonging functions for each of the linguistic terms used in the database of production rules of the fuzzy inference system should be determined for all input variables. In the process of fuzzy control of the movement of the GBMRP, both input and output linguistic variables are used. The input linguistic variables represent the data that enters the fuzzy logic system for analysis. Instead of exact numerical values, these variables can be represented by fuzzy concepts, i.e., terms. Fuzzification tools for converting input navigation data (distance d, target orientation angle φ) into linguistic variables are implemented by a microcontroller that performs data collection.

Fuzzy logic rule base for GBMRP traffic control

The formation of a base of fuzzy logic rules for managing the movement of GBMRP is a complex and multifaceted process that involves creating clear logic for real-time decision-making. The primary purpose of such management is to ensure the reliability, safety, and efficiency of the platform's movement in conditions with dynamic changes in the situation.

After detailing the functions of the input and output linguistic variables, it is necessary to develop rule bases for the two output variables, V_R, and V_L. The base of fuzzy logic rules for the output V_R variable is given in **Table 1**.

The input linguistic variable d (distance from the GBMRP to the target) is divided into five terms over the measurement interval: VSD – very small distance, SD – small distance, AD – average distance, BD – big distance, VBD – very big distance. The input linguistic variable φ (the angle of orientation to the target) is divided into seven terms in the range [−180°..+180°]: NBA – negative big angle, NAA – negative big angle, NSA – negative small angle, ZA – zero angle, PSA – positive small angle, PAA – positive average angle, PBA – positive big angle.

The base of fuzzy logic rules for the output variable V_L is given in **Table 2**.

Each fuzzy logic rule base for the output variables V_R (**Table 1**) and V_L (**Table 2**) includes 35 rules of the form *IF-THEN*. The output linguistic variables of the controller (speed of the right V_R and left wheels V_L) of the GBMRP are divided into five terms: VLS – very low speed, LS – low speed, AS – average speed, BS – big speed, VBS – very big speed.

Table 1. **Fuzzy logic rule base for the output variable V_R**

Distance, d	Angle, φ						
	NBA	NAA	NSA	ZA	PSA	PAA	PBA
VSD	BS	AS	LS	LS	VLS	VLS	VLS
SD	VBS	BS	AS	LS	VLS	VLS	VLS
AD	VBS	VBS	BS	AS	VLS	VLS	VLS
BD	VBS	VBS	VBS	BS	VLS	VLS	VLS
VBD	VBS	VBS	VBS	VBS	VLS	VLS	VLS

Table 2. **Fuzzy logic rule base for the output variable V_L**

Distance, *d*	Angle, φ						
	NBA	NAA	NSA	ZA	PSA	PAA	PBA
VSD	VLS	VLS	VLS	LS	LS	LS	AS
SD	VLS	VLS	VLS	LS	AS	BS	VBS
AD	VLS	VLS	VLS	AS	BS	VBS	VBS
BD	VLS	VLS	VLS	BS	VBS	VBS	VBS
VBD	VLS	VLS	VLS	VBS	VBS	VBS	VBS

Methods and means of forming fuzzy conclusions

The means of forming fuzzy conclusions are an important part of the system of fuzzy control of the movement of the GBMRP. The formation of a fuzzy output provides control over the movement of the GBMRP in conditions with dynamic changes in the environment's situation. The process of forming fuzzy inference is based on the rules of fuzzy logic by performing the following stages: aggregation, activation, and accumulation.

At the aggregation stage, fuzzy sets are combined that correspond to the bases of fuzzy logic rules. For each rule, the degree of execution is determined, which depends on fuzzy input values and their comparison with the conditions of the rules. Methods such as minimum (AND) and maximum (OR) functions, or other functions, are commonly used to determine the results.

At the activation stage, the degree of truth of the original variable for each rule is determined. This is achieved by applying an activation function, for example, by limiting the value of the degree of truth (truncation). The result of each rule is represented as a truncated or modified fuzzy set.

The final stage of fuzzy inference is accumulation, which involves determining the membership function for each of the original linguistic variables in the set. At this stage, the initial values of all fuzzy rules are combined into a single fuzzy multiple value, representing the system's resulting output. The following accumulation methods are used to combine fuzzy sets: Maximum, Summation, Probabilistic Sum, Average, Weighted Average, Fuzzy Integral, and Minimum. When using the maximization method, the maximum value of the belonging function is selected from all results. This method is used when it is necessary to find the most significant answer from the rules.

Methods and means of neural-like defuzzification

For defuzzification, we will utilize a neural network based on the SGTM paradigm, which assigns a numerical value to each of the initial linguistic variables.

Such neural-like defuzzification provides a reduction in computational complexity and an increase in the accuracy of receiving the GBMRP control signal.

The schemes of interaction of the fuzzy inference former with the neural network of defuzzification are shown in **Fig. 2**, where $x_1,...x_n$ – input control variables; $F_{y1},...F_{yn}$ is the output parameter of the fuzzy inference former, the number of which corresponds to the number of terms of the original linguistic variable; y *is* the resulting value of the control signal.

The architecture of a neural-like defuzzifier based on the SGTM paradigm, which determines the numerical value of the original variable δ, is illustrated in **Fig. 3**.

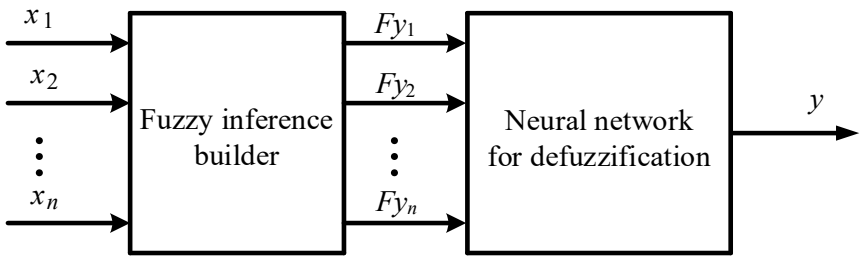


Fig. 2. Schemes of interaction of the fuzzy inference generator with a neural network for defuzzification/

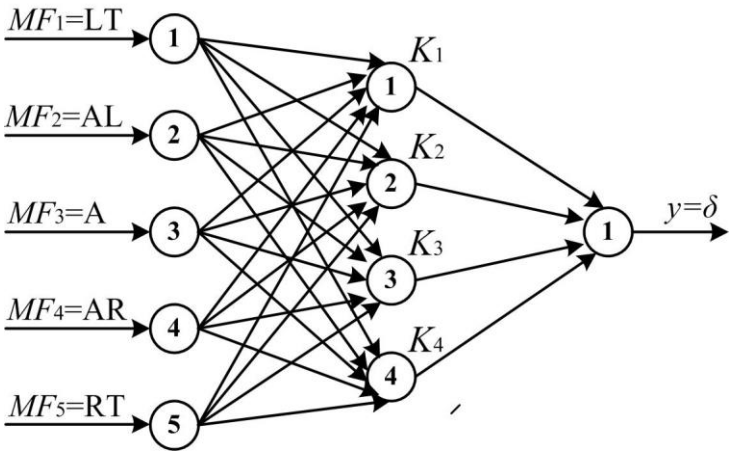


Fig. 3. Architecture of a neural-like defuzzifier for finding the numerical value of the output variable δ.

A neural defuzzifier is used in the fuzzy logic controller to control the movement of the GBMRP based on fuzzy logic. The following input and output linguistic variables are used for such control:

- Distance (D) is the distance from the GBMRP to the object (input variable);
- Angle (θ) is the angle between the object and the input variable.
- Deviation (δ) is the deviation of GBMRP relative to the input data *D* and *θ* (output variable).

The affiliation function $MF_\delta(x)$ for the output variable (δ) is given by a triangular affiliation function with the following five terms: T(δ)={LT, AL, A, AR, AT}.

The form of the membership function $MF_\delta(x)$ for the original variable (δ) is shown in **Fig. 4**.

The ownership function $MF_\delta(x)$ of the original variable (δ) is determined by the parameters a, b, c. These parameters are determined based on experimental or expert data. Its value at point x is calculated by the expression:

$$MF_\delta(x) = \begin{cases} 1 - \dfrac{b-x}{b-a}; & a \le x \le b \\ 1 - \dfrac{x-b}{c-b}; & b \le x \le c \\ 0; & x \notin (a,c) \end{cases} \quad (4)$$

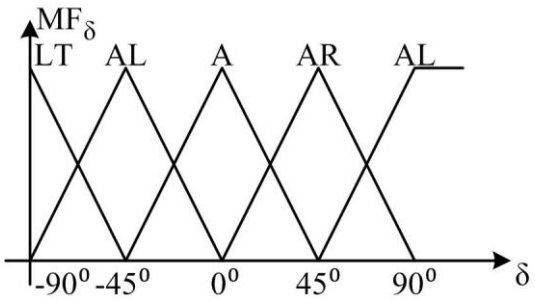


Fig. 4. Appearance of the membership function of the output variable δ.

For the output *of* $y = \delta$ of the neuro-like defuzzifier, a table of training vectors was formed based on the belonging function (**Table 3**).

Table 3. **Table of training vectors for output $y = \delta$**

Number	MF1	MF2	MF3	MF4	MF5	δ (deg)
1	0.533	0.467	0	0	0	-69
2	0.578	0.422	0	0	0	-71
3	0.378	0.622	0	0	0	-62
4	0	0.333	0.667	0	0	-15
5	0	0.289	0.711	0	0	-13
6	0	0	0.756	0.244	0	11
7	0	0	0.800	0.200	0	9
8	0	0	0.156	0.844	0	38
9	0	0	0	0.889	0.111	50
10	0	0	0	0.933	0.067	48
11	0	0	0	0.044	0.956	88
12	0	0	0	0	1.0	90

The stage of neural defuzzification based on fuzzy logic provides the receipt of a signal $y = \delta$ to control the movement of the GBMRP. In the process of learning, vectors are calculated: $M_j^{(m)}$, $\alpha_j^{(m)}$ and β_j, where $j = 0,...,n$; $m = 1,...,n-1$, n is the number of neurons of the input layer of the network. The results obtained are recorded in files and used in the operation of the neural defuzzifier in the mode of operation.

Tabular-algorithmic method of increasing the speed of means of neurofuzzy control of the GBMRP

When implementing the means of neurofuzzy control by the GBMRP, the problem of providing a real-time mode, reducing weight, dimensions, power consumption, and cost arise. One way to ensure such high technical and economic characteristics is the use of the tabular-algorithmic method for implementing the basic operations of algorithms for neurofuzzy control within the GBMRP. Most of the basic operations of the algorithms for neurofuzzy control of the GBMRP are reduced to calculating the scalar product. When implementing the operation of calculating the scalar product, it is advisable to use a multi-operand approach, in which the process of calculating the scalar product is considered as

the execution of a single operation based on elementary arithmetic operations. The basis of this approach is multiplication algorithms with the direct formation of partial products starting from the lowest digits of the factors.

The calculation of the scalar product in floating-point format is written as follows:

$$Z = \sum_{s=1}^{k} W_s X_s = \sum_{s=1}^{k} w_s 2^{m_{W_s}} x_s 2^{m_{xs}}, \quad (5)$$

where w_s and m_{W_s} – respectively the mantissa and the order of the multiplied W_s; x_s and m_{xs} – respectively the mantissa and the order of the input given (multiplier) X_s; k is the number of pairs of products, $s = 1,..,k$. To calculate the scalar product with the previously known multiplication of W_s , we will use the tabular-algorithmic method, which involves the preliminary calculation of tables of macropartial products of mantissa p_{Mi} and the determination of the maximum value of the order of macropartial products $max\, m_{P_{M_i}}$, where n is the bit depth of the mantissa of factors x_s, $i = 0,...,n$.

Kinematic model of GBMRP

Based on the geometric and mechanical properties of the GBMRP, a differentially controlled kinematic model is developed. For example, the GBMRP may use one passive and two independent active wheels for movement, which are driven by DC motors. The passive wheel of the GBMRP is used to maintain the balance of the platform. Such a wheel does not generate a traction force, but rather follows the movement of the platform. The active wheels are positioned symmetrically relative to the central axis of the platform and are independently controlled by motors, ensuring the rotation of the wheels at specified speeds. A key feature of such a kinematic model is that the passive wheel does not directly affect the model; instead, it ensures the speeds of the active wheels entirely determine the stability of the GBMRP and the movement of the platform. A kinematic model of the GBMRP is used to plan trajectories, control, and simulate the movement of the differential drive platform.

Discussion of research results

The developed information technology for autonomous neurofuzzy control of the GBMRP movement is effective in ensuring the autonomy, adaptability, and stability of the system in response to dynamic environmental changes and external interference. The conducted experiments and simulations confirmed that the combination of fuzzy logic and neuro-like structures, based on the Sequential Geometric Transformation Model paradigm, provides an increase in the accuracy of movement without the need to create an accurate mathematical model of the GBMRP and its environment. Compared to classical control methods [25, 26], which require precise mathematical models and do not adapt well to uncertainty, the proposed neurofuzzy technology demonstrates:

- greater resistance to noise and incompleteness of sensor data;
- high adaptability to dynamic changes in the environment;
- smoothness and safety of GBMRP maneuvers;
- the ability to implement on hardware and software with limited resources.

The results obtained indicate that the proposed information technology for autonomous neurofuzzy movement control of GBMRP is effective, reliable, and suitable for practical implementation. The combination of fuzzy logic, neural-like structures, and a component-oriented approach provides the system with high technical and operational characteristics, confirming the relevance and expediency of using this technology for the autonomous control of robotic platform movement in complex, changeable, and variable conditions. Therefore,

based on the results of the work performed, it is possible to formulate the following scientific novelty and practical significance of the research results. The scientific novelty of the obtained research results is the development of an information technology for neurofuzzy control of the movement of the GBMRP, which, due to the use of methods and means of collecting, storing and processing navigation data from sensors in the conditions of interference and incompleteness of information, a combination of fuzzy logic and neuro-like structures, triangular membership functions for fuzzification of input navigation data, a database of fuzzy logic rules, neural defuzzification based on SGTM provided adaptability and decision-making under uncertainty, high speed and high accuracy of real-time control signal generation.

The practical significance of the research results lies in the creation and implementation of the neurofuzzy information technology for GBMRP control, which enhances the speed, accuracy, and stability of functioning in uncertain and dynamic environmental conditions. The use of the developed methods, models and components of the information technology of neurofuzzy motion control of the GBMRP provides: adaptability of the control system to changes in traffic parameters and road environment; collision avoidance thanks to the integration of intelligent navigation algorithms and sensor data; real-time management through the use of tabular-algorithmic methods for the implementation of basic operations of neural defuzzification and navigation data processing.

CONCLUSION

It is proposed to develop an information technology for autonomous neurofuzzy control of the movement of the GBMRP in conditions with a dynamic change of the situation based on an integrated approach, which includes methods and means of collecting, storing, and processing navigation data, planning the route of the GBMRP, fuzzy logic, neural-like SGTM-based networks, a tabular-algorithmic method for calculating basic operations, and a modern element base. It is proposed to implement information technology means for neurofuzzy control of the GBMRP movement based on a problem-oriented approach, which involves a combination of software and specialized hardware. The following principles have been chosen for the implementation of the means of information technology for controlling the movement of the GBMRP: variability of the equipment composition, which provides the presence of a processor core and replaceable specialized tools; modularity; pipeline and spatial parallelism of data processing; openness of software; specialization and adaptation of hardware to the structure of data processing algorithms. An information technology for neurofuzzy control of the movement of the GBMRP has been developed, which, due to the use of methods and means of collecting, storing and processing navigation data from sensors under the influence of interference and incompleteness of information, a combination of fuzzy logic and neural structures, neural defuzzification based on SGTM provided adaptability and decision-making in conditions of uncertainty, high speed and high accuracy of real-time control signal generation. It is demonstrated that the combination of fuzzy logic and a neural-like network based on the SGTM offers adaptability and decision-making capabilities under uncertainty, as well as neuro-like defuzzification with high speed and accuracy.

It is shown that the use of triangular affiliation functions for fuzzification of input navigation data (distance to target d and angle of orientation to target φ) provided a smooth transition between levels of fuzzy sets, reduced the impact of noise and sudden changes in sensor signals, which is confirmed by the stable operation of the controller in various traffic scenarios. A set of rules for fuzzy logic has been developed, which provides an adequate representation of expert knowledge in the control system, as evidenced by the correct adaptation of wheel speeds when changing the distance and angle of orientation of the platform. The effectiveness of the rule base is confirmed by the smoothness of the

GBMRP's maneuvers, as well as the absence of sudden accelerations and braking, even in unpredictable environmental changes.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was performed within the R&D "Methods and means of intelligent measurement of movement parameters and determination of spatial orientation of ground mobile robotic platforms" [0124U000822] carried out by Lviv Polytechnic National University and funded from the state budget of the Ministry of Education and Science of Ukraine for 2024-2026.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [I.Ts., V.T.]; methodology, [I.Ts., V.T.]; investigation, [Yu.O., V.R.]; writing – original draft preparation, [I.Ts., V.T.]; writing – review and editing, [Yu.O.]; visualization, [O.O.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Al Mahmud, S., Kamarulariffin, A., & Ibrahim, A. M. (2024). Advancements and challenges in mobile robot navigation: A comprehensive review of algorithms and potential for self-learning approaches. Journal of Intelligent & Robotic Systems, 110(1), 120. https://doi.org/10.1007/s10846-024-02149-5

[2] Al-Mayyahi, A., & Al-Mansoori, M. (2023). Adaptive neuro-fuzzy inference system for autonomous ground vehicle navigation. Journal of Robotics and Automation, 12(3), 45–59. https://doi.org/10.1016/j.jra.2023.03.005

[3] Antonelli, G., & Siciliano, B. (2022). Fuzzy logic-based path-following for differential drive mobile robots. IEEE Transactions on Robotics and Automation, 18(4), 564–573. https://doi.org/10.1109/TRA.2022.3145678

[4] Baturone, I., & García, J. (2023). Low-cost embedded neuro-fuzzy controller for mobile robot navigation. Robotics and Autonomous Systems, 145, 103652. https://doi.org/10.1016/j.robot.2023.103652

[5] Boubertakh, M., & Boudjema, F. (2021). Design of fuzzy controllers for obstacle avoidance and goal-seeking behaviors in mobile robots. International Journal of Intelligent Robotics and Applications, 5(2), 123–135. https://doi.org/10.1007/s41315-021-00110-5

[6] Brahmi, M., & Boukhetala, D. (2020). Path planning and localization of mobile robots using recurrent neural networks. Neural Computing and Applications, 32(7), 1955–1967. https://doi.org/10.1007/s00542-020-05583-3

[7] Demirli, E., & Khoshnejad, M. (2019). Sensor-based neuro-fuzzy controller for autonomous parallel parking of a car-like mobile robot. Journal of Intelligent & Robotic Systems, 95(2), 345–358. https://doi.org/10.1007/s10846-019-01052-3

[8] Dongshu, L., & Zhang, X. (2018). Intelligent autonomous navigation structure for a vehicle using multi-layered neural networks. Journal of Intelligent Systems, 27(1), 45–56. https://doi.org/10.1515/jisys-2017-0051

[9] Deshpande, S., & Bhosale, S. (2017). Navigation of a nonholonomic wheeled mobile robot using ANFIS controller. Procedia Computer Science, 115, 319–326. https://doi.org/10.1016/j.procs.2017.09.139

[10] Bodiansky E. V. et al. Analysis and processing of data flows by means of computational intelligence: Monograph – Lviv: Lviv Polytechnic Publishing House, 2016. – 235 pp. (In Ukrainian).
[11] Godjevac. J. (1999) Neuro-fuzzy control of a mobile robot, Proc. 1999 IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan, pp. 145–150.
[12] Gharajeh, M. S., Nadimi-Shahraki, M. H. and Gharajeh, M. E. (2020) Hybrid GPS-ANFIS approach for collision-free navigation of mobile robots, Robotics and Autonomous Systems, vol. 133. https://doi.org/10.1016/j.robot.2020.103669
[13] Hilali, B. (2023) Neuro-Fuzzy Combination for Reactive Mobile Robot Navigation, Artificial Intelligence Advances, vol. 3, no. 2. https://doi.org/10.52549/ijeei.v11i2.4009
[14] Al-Mahmud, H. H. and Islam, M. M. (2024) Advancements and Challenges in Mobile Robot Navigation: A Comprehensive Review," Applied Sciences, vol. 14, no. 1, pp. 1–27. https://doi.org/10.1007/s10846-024-02149-5
[15] Dey, S. R., Hossain, A. and Rahman, F. (2025) Deep Reinforcement Learning-Based ANFIS Controller for Autonomous Navigation, IEEE Access, vol. 13, pp. 18763–18778.
[16] V. M. Khoma, "Model of sequential geometric transformations in neuro-like structures," Bulletin of the National University "Lviv Polytechnic",№ 879, c. 112–121, 2017. (In Ukrainian).
[17] O. V. Lutsyk, "Application of the model of successive geometric transformations in neurofuzzy control systems," Control, Navigation and Communication Systems, №2(64), c. 38–45, 2022. (In Ukrainian).
[18] A. Mahmud et al., (2024) Real-Time Implementation of Neuro-Fuzzy Control Systems on FPGA for Mobile Robots, IEEE Transactions on Industrial Electronics, vol. 71, no. 9, pp. 9876–9887. https://doi.org/10.14569/IJACSA.2016.070136
[19] Elshamli, M., & Al-Mahmoud, A. (2016). Genetic algorithm technique for solving the path planning problem of a mobile robot in static and dynamic environments. International Journal of Advanced Robotic Systems, 13(1), 1–12. https://doi.org/10.5772/62052
[20] Elamvazuthi, I., & Rashid, R. (2015). Fuzzy-based navigation and control of a non-holonomic mobile robot. International Journal of Computer Applications, 118(24), 1–6. https://doi.org/10.5120/20809-9444
[21] Elkady, M., & Elhoseny, M. (2024). Adaptive neuro-fuzzy inference system for mobile robot navigation in dynamic environments. Computers, Materials & Continua, 70(2), 1773–1788. https://doi.org/10.32604/cmc.2024.027883
[22] Rashkevych, Yu.M., Tkachenko, R.O., Tsmots, I.G., Peleshko, D.D. Neural-like methods, algorithms and structures for real-time signal and image processing: Monograph – Lviv:2010.-130c. (In Ukrainian).
[23] García, J., & Baturone, I. (2020). Low-cost embedded neuro-fuzzy controller for mobile robot navigation. Robotics and Autonomous Systems, 144, 103652. https://doi.org/10.1016/j.robot.2020.103652
[24] Gharajeh, M. S., & Gharajeh, M. S. (2022). An intelligent approach for autonomous mobile robots path planning. Robotics and Autonomous Systems, 145, 103652. https://doi.org/10.1016/j.robot.2022.103652.
[25] Dong, L., He, Z., Song, C., & Sun, C. (2021). A review of mobile robot motion planning methods: From classical motion planning workflows to reinforcement learning-based architectures. IEEE Access, 9, 104425–104441. https://doi.org/10.1109/ACCESS.2021.3096763
[26] Liu, L., Zhang, J., & Wang, Z. (2023). Path planning techniques for mobile robots: Review and future directions. Journal of Intelligent & Robotic Systems, 101(1), 1–26. https://doi.org/10.1007/s10846-022-01615-5}

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОНОМНОГО НЕЙРОНЕЧІТКОГО УПРАВЛІННЯ РУХОМ НАЗЕМНОЇ МОБІЛЬНОЇ РОБОТОТЕХНІЧНОЇ ПЛАТФОРМИ

Іван Цмоць[1]*, Василь Теслюк[1], Юрій Опотяк[1], Василь Рабик[2], Олександр Олійник[1]
[1]Національний університет «Львівська політехніка», вул. Степана Бандери 12, м. Львів, 79013 Україна
[2]Львівський національний університет імені Івана Франка, вул. Університетська, 1, м. Львів, 79000, Україна

АНОТАЦІЯ

Вступ. Сучасний етап розвитку наземних мобільних роботизованих платформ (НМРП) характеризується зростанням вимог до системи управління рухом щодо автономності, адаптивності до динамічних змін у оточуючому середовищі при врахуванні обмежень на обчислювальні ресурси, масо-габаритні параметри і енергоспоживання. Традиційні методи управління рухом НМРП, які ґрунтуються на точних математичних моделях, є неефективними в реальних динамічних умовах, де неможливо точно описати стан НМРП і середовища. Тому для управління рухом НМРП у таких умовах пропонується розроблення відповідної інформаційної технології на основі нечіткої логіки та нейромереж.

Матеріали та методи. Інформаційна технологія автономного нейронечіткого управління рухом НМРП розробляється на базі інтегрованого, гібридного та проблемно-орієнтованого підходів з використанням компонентно-орієнтованої технології. Методи нечіткої логіки забезпечать гнучкість і стійкість системи управління до шумів, похибок вимірювань, непередбачуваних змін у навколишньому середовищі, уможливлюють інтеграцію з нейронними мережами для підвищення точності та швидкодії. Нейромережеві методи забезпечують підвищення точності навігаційних вимірювань, відновлення втрачених даних та побудову нейроподібного дефазифікатора з підвищеною точністю формування сигналів управління.

Результати. Розроблено інформаційну технологію автономного нейронечіткого управління рухом НМРП, яка за рахунок використання методів та засобів збору, збереження та опрацювання навігаційних даних в умовах дії завад і неповноти інформації, поєднання нечіткої логіки та нейроподібних структур на основі моделі послідовних геометричних перетворень, забезпечила адаптивність та прийняття рішень в умовах невизначеності, точність формування сигналів керування. Розроблено базу правил нечіткої логіки, яка забезпечила адекватне відображення експертних знань у системі управління.

Висновки. У роботі на базі інтегрованого, гібридного та проблемно-орієнтованого підходів з використанням компонентно-орієнтованої технології розроблено інформаційну технологію автономного нейронечіткого управління рухом НМРП в умовах динамічної зміни ситуації, що враховує обмеження щодо обчислювальних ресурсів та енергоспоживання.

Ключові слова: інформаційна технологія, нечітка логіка, нейроподібна мережа, наземна мобільна робототехнічна платформа, управління рухом платформи, дефазифікація

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
14 November, 2025	01 December, 2025	01 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 39–54
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.04

SUSTAINABLE OPTIMIZATION OF CONSOLIDATED DATA PROCESSING ALGORITHMS BASED ON MACHINE LEARNING AND GENETIC ALGORITHMS

Vasyl Lyashkevych
Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine

Lyashkevych, V.Y. (2025). Sustainable Optimization of Consolidated Data Processing Algorithms Based on Machine Learning and Genetic Algorithms. *Electronics and Information Technologies*, 32, 39–54. https://doi.org/10.30970/eli.32.3

ABSTRACT

Background. Automation of analytical report generation in industrial companies is gaining strategic importance due to the variety of document formats, increasing data volumes, and growing requirements for the rapid generation of multi-component analytical materials. Traditional ETL pipelines cannot cope with the complexity of modern information flows, especially when machine learning (ML), large language models (LLMs), and agent systems are integrated into the process. Due to the rapid progress of code generation and autonomous agent capability to perform complex analytical procedures, the task of automatically constructing reporting pipelines is becoming increasingly promising and scientifically sound.

Materials and Methods. An evolutionary model for constructing algorithms for processing consolidated reports based on genetic algorithms (GA) is proposed. For report generation, the algorithm defines a pipeline for constructing a visual component. The population grows as a tensor, enabling parallel evolution of a set of independent workflows. The operations are classified into four groups: ETL, ML, LLM, and VIS. The fitness function evaluates the constant length of the pipeline, the coverage of key types of operations, and their structural consistency.

Results and Discussion. Experimental results have shown that GAs rapidly evolve from random NOP-dominated structures to stable, logically consistent, and functional pipelines with a duration of 10-14 operations. The best chromosomes formed three full-fledged visual components: a predictive regression model, semantic clustering represented by embeddings, and a categorical diagram. This evolutionary pattern confirms that combined pipelines can be built automatically and adaptively, and the increase in the complexity of operations in the “meaning” space, which is the vector space of embedding, along with the development of code generation and agent architectures.

Conclusion. The proposed model demonstrates an effective mechanism for automated synthesis of multi-visual reports based on evolutionary pipelines. The method's prospects grow with the development of AI agent systems and the increase in the number of operations in the content space, which paves the way for a complete system of autonomous analytical reporting of a new generation.

Keywords: consolidated data, sustainable optimization, machine learning, genetic algorithms, LLM, data analytics

© 2025 Vasyl Lyaskevych. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

In modern information ecosystems, where industrial companies operate with large volumes of heterogeneous data, there is a growing need not only for the automation of analytical processes, but also for sustainable approaches to optimizing consolidated data (CD) processing. The sustainability of algorithms refers to their ability to remain effective as formats change, workload increases, data sources expand, and business requirements evolve [1-2]. CD processing includes extraction, transformation, semantic structuring, and information integration that requires adaptive intelligent mechanisms capable of maintaining long-term stability, minimizing costs, and ensuring scalability in dynamic environments [3].

In this context, a combination of ML methods and GAs proves to be effective. In AutoML systems, GAs are successfully used to optimise hyperparameters, select models, create and reduce pipelines, and dynamically adjust their structure [4-7]. Due to the ability of evolutionary algorithms to search for optimal configurations in large solution spaces, their integration into industrial report processing tasks looks natural and promising. They will not be able to build evolutionary-adaptive pipelines that maintain system stability, adapt to the emergence of new formats, and minimise operational costs [8].

In conditions of a global economy, industrial and commercial enterprises interact with a wide network of suppliers and customers. It generates complex and highly variable data flows. Data comes in hundreds of disparate formats such as spreadsheets, PDFs, presentations, graphical reports, scanned forms, and hybrid structures. Usually, they vary significantly in terms of language, structure, and content. Automating such processing becomes a critical task to ensure consistency and quality of corporate analytics. Current research demonstrates the effectiveness of using LLMs, vector and graph databases [9-11], multi-agent systems [12-13], ML, deep learning (DL), and reinforcement learning (RL) methods [14-16], as well as evolutionary and GAs [6-7, 17] for optimizing ETL pipelines and analytical processes.

A separate research direction involves combining LLM with vector databases to enhance the quality of semantic search, classification, and context analysis [9]. Other works emphasise the potential of graph databases in combination with multi-agent architectures for implementing multi-step logical inference [13]. Research on optimising ETL processes using ML/RL suggests the possibility of dynamically adapting pipelines, detecting anomalies, and automatically selecting transformations [16, 18].

Considerable scientific interest is also associated with the role of LLMs in performing analytical operations in the “meaning space.” The work [19] demonstrated that LLMs are capable of performing interpretive, logical, statistical, and arithmetic operations without the use of hard-coded algorithms, operating on data in the semantic (“meaning”) space. This opens opportunities for building flexible, dynamic exploratory data analysis (EDA) processes without predefined pipelines, which is especially important in conditions of a wide variety of report formats and changing requirements.

Despite these advances, industrial reporting systems face several unresolved challenges:

1. A growing variety of formats, over a hundred or more;
2. The need to simultaneously extract transactional, contextual, and semantic data;
3. High demands on accuracy, consistency, and robustness of etl processes;
4. The need to generate new reports with different types of visualisations;
5. The need for adaptive pipeline evolution in response to the dynamics of formats and business requirements demanded in the market.

A particular problem is the high cost and instability of real experiments in an industrial environment. Testing different combinations of ETL operations, ML/LLM modules, and pipeline structures requires significant resources and time, and can lead to quality losses

or failures in productive systems. Therefore, there is a need for preliminary modelling using a simulation environment that allows for evaluating the effectiveness of various configurations without their physical implementation.

In this context, GAs are a natural tool for modelling, optimizing, and adaptively evolving report processing pipelines. Their strength lies in the fact that:

1. ETL, ML, and LLM operations can be represented as genes, and the pipeline as a chromosome.
2. GA allows for minimizing the number of operations while maintaining or improving the quality of the results.
3. GAs provide the search for the optimal sequence of operations according to contextual variables and requirements such as accuracy, time, resources, quality of the final report, required visuals, and others.
4. GAs allow for simulation of the system before its actual implementation and deployment, reducing risks and costs.
5. GAs naturally adapt to the emergence of new formats and new types of operations.

As mentioned, GAs are successfully used in AutoML systems, so there is a reasonable scientific expectation that, in the context of the current task, with the extraction of transactional, contextual, and semantic data from heterogeneous reports, the application of GAs will be extremely effective. The integration of LLM approaches in the semantic space [19], multi-level storage systems which include SQL, VectorDB, and GraphDB, multi-agent architectures, ML/DL/RL modules, and GAs forms the foundation of a new paradigm - an evolutionary-adaptive system for sustainable optimization of CD processing algorithms which can automate processing of industrial reports.

Recent advances in code generation using LLMs have significantly expanded the capabilities of automated data-processing systems, enabling dynamic construction of ETL procedures, semantic extraction modules, and visualization logic directly from natural-language specifications. State-of-the-art models such as Codex, CodeT5, GPT-4 and Llama-3 have demonstrated near-human performance in generating reusable and semantically consistent code components for data analytics workflows [20-22]. Scientific studies show that LLM-driven code generation not only accelerates pipeline development but also improves modularity, reproducibility, and correctness through learned structural patterns and contextual reasoning [23]. When applied to automated reporting, code-generating models create flexible and adaptive procedures for parsing, embedding, aggregating, storing, and visualizing data, enabling rapid reconfiguration of workflows in response to new report formats or analytical requirements. Thus, now it no longer makes sense to program all technological operations in advance and after serving them.

Simultaneously, agentic AI systems have emerged as a powerful paradigm for orchestrating multi-step analytical processes. Research demonstrates that multi-agent architectures, comprising planning agents, reasoning agents, tool-using agents, and code-executing agents, achieve superior performance in complex, multi-stage tasks such as document analysis, information synthesis, and dynamic pipeline construction [24-25]. Agentic systems benefit from autonomous task decomposition and iterative refinement, allowing agents to negotiate, supervise, and correct one another, thereby reducing error propagation and increasing robustness. When applied to automated report generation, agentic AI can coordinate the entire lifecycle: ingesting documents, selecting transformations, generating executable code, validating results, and producing narrative and visual outputs.

Before constructing a fully autonomous agentic pipeline, it is essential to first identify the optimal algorithmic structure for the report-building workflow. Agentic systems rely on a predefined space of tools and capabilities. If this space is suboptimal, redundant, or contains low-quality operations, the agents will form inefficient or even failed workflows.

Scientific evidence shows that agent-based systems are highly sensitive to the structure of available operations, often suffering from combinatorial explosion and suboptimal planning when the action space is not pre-optimized [24]. Therefore, determining the optimal algorithm, through GAs, evolutionary search, or analytical design, ensures that the agentic system operates within a validated, minimal, high-quality set of operations, maximizing performance and sustainability. Only after this optimization step, agentic AI can reliably construct scalable, interpretable, and efficient multi-visual report generation pipelines.

MATERIALS AND METHODS

Modern industrial companies support reports from numerous suppliers and customers in hundreds of disparate formats that differ in structure, data types, language, context, and semantic relationships. Those reports contain different types of CD, such as transactional, contextual, and semantic. This needs to be properly extracted, transformed, and integrated into the appropriate SQL, VectorDB, and GraphDB repositories, respectively. The growth in the number of formats, the dynamic nature of changing business requirements, and the need for scalability create a demand for continuous, sustainable optimization of CD processing processes.

In real industrial conditions, testing all possible combinations of ETL, ML, and LLM operations is extremely expensive, risky, and unsustainable: it requires significant time and computing resources, can disrupt productive processes, and lead to data quality losses. Therefore, the primary task is to create a simulation model that allows for virtually exploring and optimising data processing pipelines before their physical deployment. The most suitable tool for such optimization is GAs. Thanks to the evolutionary operators of selection, crossover, and mutation, GAs can effectively explore a large space of possible processing configurations and find stable and minimally complex pipeline variants.

Formalization of the problem

The data processing and analytics pipeline is shown in **Fig. 1**. Users can provide input reports in various formats, such as PDF, XLSX, databases, and APIs containing raw data to the pipeline. An algorithmic workflow processes the data through multiple stages: O_1, O_i , O_j and O_l , extracting insights and transforming it. The system integrates data from multiple storage systems, including SQL databases, VectorDBs, and GraphDBs, ensuring compatibility with various data formats.

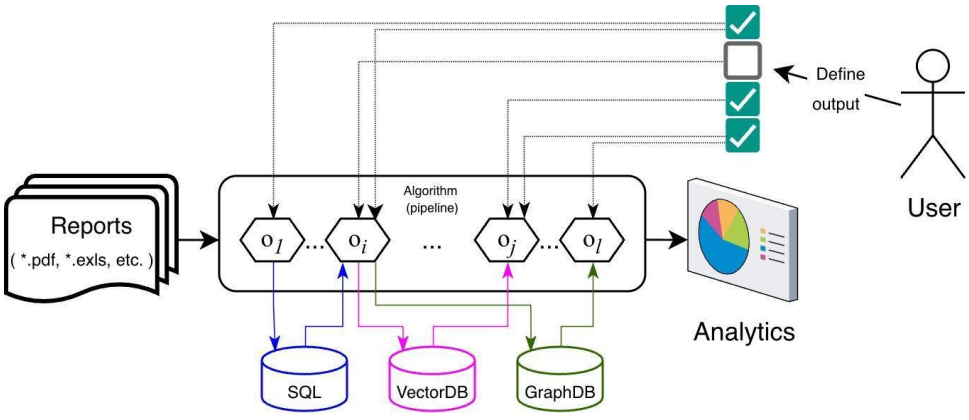


Fig. 1. Report generation approach.

A user defines the desired outputs as visual components, which determine the analysis scope, metrics, and visualizations in the report. Based on these preferences, the

processed data is transformed into tailored analytics outputs, such as visual charts or custom reports.

This architecture supports flexible processing, allowing dynamic user-driven customization of outputs and efficient multi-database integration. It is particularly useful for applications where decision-making relies on consolidated, processed insights from diverse and complex datasets and for automation based on an Agentic AI or multi-agent approaches.

Let the industrial report processing system be constructed as an evolutionary-adaptive algorithm A, consisting of an ordered set of operations:

$$A = \{o_1, o_2, \ldots, o_l\}, \; o_i \in O, \tag{1}$$

where O is the universal set of all possible operations (ETL, ML, LLM, Visualization).

Each operation is defined as:

$$o_i = (t_i, \theta_i, D_i), \tag{2}$$

where: t_i is type of operations (ETL, ML, LLM, Visualization); θ_i – parameters of operations; $D_i = \subseteq O \times O$ – a set of dependencies (pre- and post-conditions).

There are three main types of dependencies in CD processing: structural, semantic, and functional. In case of structural dependencies, an operation o_j may only be executed after the other operation o_i, for example $o_i \preccurlyeq o_j$. Formally:

$$\forall o_j \in A: Pre(o_j) \subseteq A. \tag{3}$$

The operation o_j requires semantic entities produced earlier:

$$Entities(o_j) \subseteq \cup_{i<j} Output(o_i), \tag{4}$$

Example: “LLM summarization” depends on “context normalization”, which depends on “Entity extraction”.

Operation o_j may require a specific type of input:

$$TypeIn(o_j) = TypeOut(o_i). \tag{5}$$

Example: “ML classification” requires embeddings produced by a “vectorization operation”.

Therefore, a CD processing algorithm A is valid when the conditions (3), (4), and (5) are met simultaneously. Thus, GA evolves not arbitrary sequences but structurally valid algorithms.

The context of a document $c \in C$ constrains which operations may be applied:

$$O_c = \{o \in O \mid Applicable(o, c) = 1\}. \tag{6}$$

Example of context-dependent operations:

- if a document is in Chinese → translation “+” multilingual normalization required;
- If PDF → OCR and structural extraction are required;
- If it contains graphical tables → table reconstruction is needed.

Thus: $A_c \subseteq O_c$.

Data transformation through the algorithm A is defined as:

$$X_0 = r \in R; \quad X_{i+1} = o_i(X_i); \quad X_l = A(r). \tag{7}$$

Required outputs: transactional data $T(r)$, contextual embeddings $V(r)$, and semantic graph $G(r)$. Algorithm correctness condition $T(r), V(r), G(r) \neq \emptyset$:.

The optimization objective of the fitness function with dependency penalties is:

$$F(A) = \beta_1 E(A) + \beta_2 L(A) - \beta_3 Q(A) - \beta_4 S(A) + \beta_5 D(A), \tag{8}$$

where: $E(A)$ is number of operations, $L(A)$ – latency/compute cost, $Q(A)$ – extraction quality such as accuracy, recall, and consistency, $S(A)$ – sustainability/adaptiveness, $D(A)$ – dependency violation penalty. Penalty:

$$D(A) = \lambda \cdot |\{o_i \in A \mid not\ Valid(o_i)\}|, \tag{9}$$

If the algorithm satisfies all dependencies, it will be 1, otherwise 0. Therefore, algorithm validity, using (3), (4), and (5), is being calculated:

$$Valid(A) = \wedge_{j=1}^{l}\left[Pre(o_j) \subseteq \{o_1, \ldots, o_{j-1}\}\right] \wedge \left[TypeIn(o_j) = TypeOut(o_{j-1})\right]. \tag{10}$$

Finally, the optimization objective:

$$A* = arg\ min_{P \in P_{valid}} F(A). \tag{11}$$

The goal is to find and evolutionarily optimize a data-processing algorithm, which emulates a report processing pipeline, that satisfies all structural, functional, and semantic dependencies between operations while minimising the number of operations and computational cost, maximizing the extraction quality of transactional, contextual, and semantic data, and maintaining adaptability to new report formats.

Sets of operations for experiments

Defining sets of ETL, ML, LLM, and visualization operations is essential for building a formal, managed, and optimized report processing process. When a system receives heterogeneous reports, it must know what transformations to perform: what data to parse, what embeddings to create, how to aggregate, and where to store the information. A clear separation of operations allows building scalable pipelines that support three data types: transactional (SQL), contextual (VectorDB), and semantic (GraphDB). It also provides automatic selection of ML, LLM, and visualization components and optimization of their sequence using GAs. Examples of operations are shown in **Table 1**.

The global operation space can be described as:

$$O = O_{ETL} \cup O_{ML} \cup O_{LLM} \cup O_{VIS}, \tag{12}$$

where: O_{ETL} are operations for loading, converting, merging and storing data; O_{ML} – operations for statistical and machine learning; O_{LLM} – semantic, linguistic and reasoning operations; O_{VIS} – operations for building reports and visualizations.

Combined operation space is a unified space of all possible operations that a system can use for automated report processing. It combines four large classes: ETL, ML, LLM, and Visualization. Each operation performs a specific function such as parsing, cleansing, aggregation, classification, semantic extraction, text generation, or visualization of results.

Table 1. **Sets of used operations in experiments**

Operation	Functionality	Samples
O_{ETL}^{ing}	Input, upload, and ingestion	UploadFile, DownloadFromAPI, IngestEmail, FetchFTP
O_{ETL}^{parse}	Parsing and extraction	PDFParse, OCR, TableDetection, ExcelParse, CSVParse, ImageToTable, HTMLParse
O_{ETL}^{norm}	Data normalization	CleanText, NormalizeNumbers, Deduplicate, FixEncoding, DateNormalization, CurrencyNormalization
O_{ETL}^{agg}	Data aggregation and fusion	JoinRecords, MergeSources, GroupBy, Summarize, Pivot, Unpivot
O_{ETL}^{store}	Data storing	StoreSQL(T), StoreVectorDB(V), StoreGraphDB(G)
O_{ETL}^{read}	Data reading	ReadSQL, ReadVector, ReadGraph
O_{ML}^{clas}	ML classification	DocumentTypeClassifier, AnomalyClassifier, SupplierCategoryClassifier
O_{ML}^{reg}	ML regression	PredictMissingValues, ForecastQuantities, CostRegression, DeliveryTimeRegression
O_{ML}^{clust}	ML clustering	ClusterReports, ClusterSuppliers, ClusterContentTopics
O_{ML}^{emb}	Embedding-based ML	ComputeEmbeddings, SemanticSimilarity, NNRetrieval
O_{LLM}^{reason}	Semantic understanding and reasoning	SemanticParsing, LogicalInference, MultiHopReasoning, ConstraintValidation
O_{LLM}^{gen}	Summarization and generation	ShortSummary, DetailedSummary, ExecutiveReport, NarrativeGeneration
$O_{LLM}^{extract}$	Information extraction	EntityExtraction, RelationExtraction, AttributeExtraction, SchemaInduction
O_{LLM}^{trans}	Transformation	Translate, NormalizeTerms, Rewrite, CleanNoise, ContextualRewriting
O_{LLM}^{etl}	LLM for ETL	LLM-TableReconstruction, LLM-DataValidation, LLM-SQLQueryGeneration
O_{VIS}^{basic}	Basic charts	BarChart, LineChart, PieChart, Histogram, ScatterPlot
O_{VIS}^{adv}	Advanced analytics visuals	Heatmap, Sankey, GeoMap, TimeSeriesForecastPlot, CorrelationMatrix
O_{VIS}^{narr}	Narrative visualization	ChartExplanation, NarrativeAnalytics, LLM-GeneratedReport
NOP	No Operation	NOP

Combining these sets into a single space O allows:

- to consider any operation as a gene in the pipeline chromosome;
- to build flexible, adaptive, and diverse pipelines;
- allow the GA to freely combine and optimize different types of operations;
- provide support for all data types: transactional (SQL), contextual (VectorDB), and semantic (GraphDB).

Thus, the combined operation space is a foundation that utilizes all the transformation capabilities that the system can perform on data. Thus, for example, the combined operation space can be defined:

$$O = O_{ETL}^{ing} \cup O_{ETL}^{parse} \cup O_{ETL}^{norm} \cup O_{ETL}^{agg} \cup O_{ETL}^{store} \cup O_{ML} \cup O_{LLM} \cup O_{VIS}, \quad (13)$$

Every element O can be gene in a chromosome, forming an executable pipeline.

General model of operation

Each operation should be formalized within a unified operational specification suitable for LLM, ML, ETL, and Visualization classes. Thus, an operation is described as:

$$o = (id, type, X_{in}, X_{out}, \theta, C, Dep), \quad (14)$$

where: id is unique transaction identifier; $type$ – class of ETL, ML, LLM or VIS; X_{in} – type of input data; X_{out} – type of output data; θ – operation parameters; C – contextual execution conditions and Dep – depending on other operations.

Prepare GA for experiments

Encoding operations and chromosome structure. Each operation from the combined operation space (ETL, ML, LLM, VIS) is encoded as a gene, represented by a unique integer or symbolic token:

$$o_i \rightarrow g_i \in \Sigma, \quad (15)$$

where Σ is the operation alphabet.

In the classical formulation of evolutionary pipelines, a chromosome is represented as a single linear sequence of operations:

$$P = [g_1, g_2, \dots, g_L], \quad (16)$$

where each gene corresponds to a transformation step in a pipeline.

Because different pipelines may require different numbers of operations, we use a fixed-length chromosome. A fixed length L guarantees comparability among individuals and stable crossover or mutation behaviour. To allow variable-length logic within a fixed-length chromosome, unused positions are filled with a NOP gene:

$$NOP(X) = X. \quad (17)$$

The NOP operator allows the GA to simulate shorter pipelines within a fixed-length representation, supports incremental growth of solutions, and prevents destructive crossover.

However, this representation encodes only one computational workflow, suitable for generating a single analytical output (one visualization or one analytical component). In the context of automated report generation, a user typically requires multiple analytical outputs simultaneously, such as n different charts, KPIs, summaries, embeddings, graphs or others. Each of these outputs must be built by a distinct computational pipeline, possibly sharing some operations but often diverging after early stages. Therefore, a single linear chromosome is insufficient.

Typical document-processing pipelines include from 20 to 50 operations, depending on operational diversity. A single pipeline builds only one analytical output, but real reports

require multiple independent visuals. In this case, the chromosome must be represented as a matrix of size $L \times n$. Consequently, the GA population (**Fig. 2**) becomes a tensor $M \times L \times n$, enabling simultaneous evolution of multiple pipelines under shared constraints:

$$P \in R^{L \times n}, \tag{18}$$

where: L is max pipeline length (50 operations per one visual or analytical component), n – number of requested visuals or analytical components, each column $P_{i,j}$ encodes one full algorithmic pipeline for visual j.

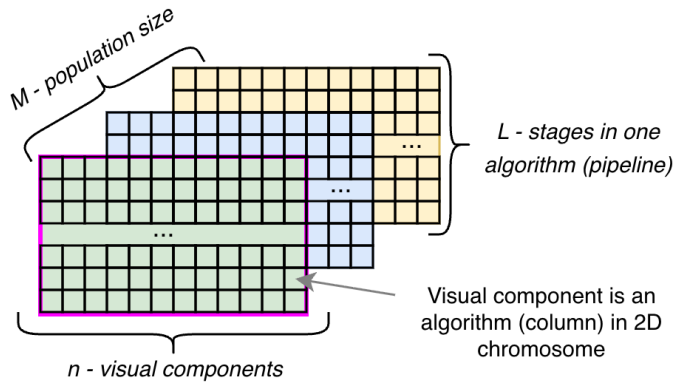


Fig. 2. Structure of a chromosome.

With matrix chromosomes, the population is no longer a 2-D matrix (population size × genome length). It becomes a 3-dimensional tensor:

$$P \in R^{M \times L \times n}, \tag{19}$$

where: M is population size, L – pipeline length per output, n – number of independent visuals.

The proposed chromosome's structure considers:

- Multi-objective report generation. A report with n visuals is not a single-objective optimization problem but a multi-pipeline multi-objective system. Each visual requires a different data-processing workflow.
- Preservation of algorithm (pipeline) independence. Encoding all visuals in a single vector would mix operations across visuals and destroy semantic locality. Matrix representation preserves the independence of pipelines.
- Parallel evolution in a shared context. Early operations such as parsing, OCR, and aggregations may be shared, while later stages such as special aggregations, ML, LLM, or visualization diverge. Matrix form allows partial sharing naturally.
- Tensor representation enables proper genetic operators. Mutation and crossover can be applied, such as column-wise (per visual), row-wise (per pipeline depth), and block-wise (submatrix crossover).
- Scalable and sustainable optimization. Industrial reports frequently contain 10-20 visuals, avoiding redundant computation and ensuring consistency.

The initial population is created using heuristic seeding. We assume that a subset of 10-20% of the population is initialized using heuristically valid partial pipelines:

- parsing → normalization → embedding → ML/LLM → storage;

- OCR → table reconstruction → aggregation → storage;
- parsing → LLM extraction → graph building.

This accelerates convergence.

Some individuals are initialized with many NOP genes and a small number of operations inserted in valid positions. This encourages smooth evolutionary growth from simple to complex solutions.

RESULTS AND DISCUSSION

The proposed GA was evaluated through controlled simulations designed to model the automated construction of multi-visual analytical reports. Due to the limited quantity of the prepared operations, we cannot allow the use of a large population. Thus, the experiment used a population of 40 matrix-encoded chromosomes, each representing a set of three parallel pipelines (one per visual) with a maximum depth $L = 30$ of operations. Thus, each chromosome formed a tensor-shaped individual of size 30×3, while the entire population spanned a tensor of size $40 \times 30 \times 3$. Such representation enabled the simultaneous evolution of multiple workflows required for multi-visual reporting, which cannot be encoded in a classical one-dimensional chromosome.

Fitness function evaluation

The GA was executed for 50 generations. Fitness was calculated per visual using a function that rewarded: sustainable pipeline length (~10 operations), presence of ETL, ML, LLM, and VIS categories, and overall completeness of the workflow.

For each individual, the fitness is computed per visual (per column) and then summed across visuals:

$$F(P) = v = \sum_{v=1}^{N_{vis}} f(P_v). \tag{20}$$

Example of calculations for one visual component:

$$f = -|\, effective_{length} - 10\,| + 3 \cdot has_{ETL} + \\ +2 \cdot has_{ML} + 2 \cdot has\_LLM + 4 \cdot has\_VIS.$$

As is seen from the provided example, we expect approximately 10 operations in the pipeline as an effective length. GA optimization dynamics (**Fig. 3**) show rapid improvement during the first 10 generations, followed by gradual stabilization.

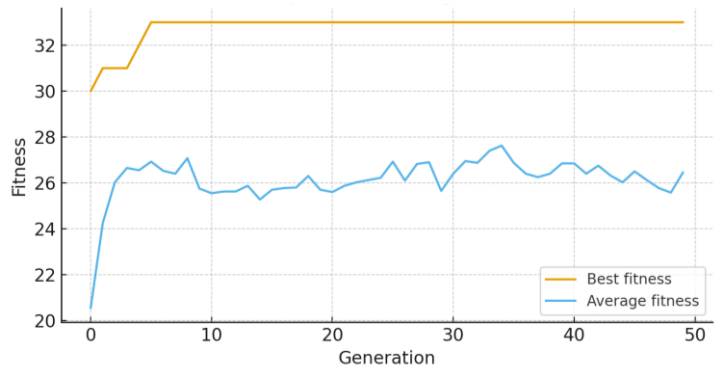


Fig. 3. Fitness function estimation.

The best fitness converged at 33, while the population mean stabilised around 26. This indicates that the GA successfully moved from sparse, primarily NOP-heavy pipelines toward meaningful multi-stage workflows, combining ETL, ML, LLM, and VIS operations.

Pipeline length distribution

Fig. 4 shows the distribution of effective pipeline lengths for the final generation. The population converged around 10-14 operations per pipeline, which aligns with the sustainability objective encoded into the fitness function.

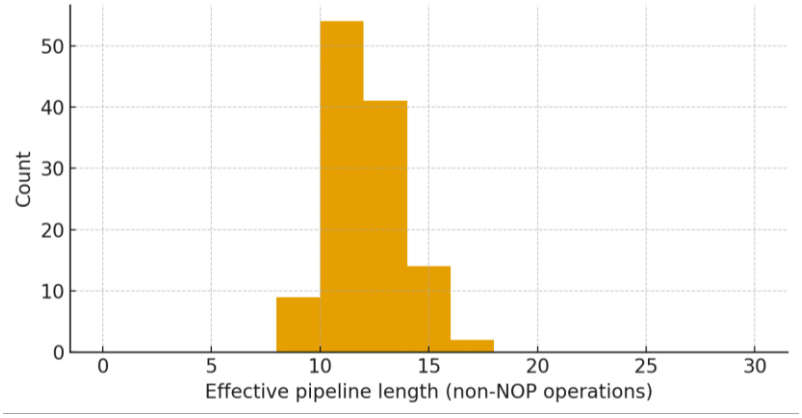


Fig. 4. Pipeline length distribution.

Pipelines shorter than 6 steps lacked analytic capability, while pipelines longer than 15 steps accumulated penalties due to inefficiency. The GA thus demonstrated an emergent preference for compact, interpretable, resource-efficient pipelines.

Pipeline decoding

The best-performing chromosome consisted of a 30×3 matrix, in which three distinct pipelines were evolved. Each pipeline included a mixture of:

- ETL steps: parsing, normalization, and table extraction;
- ML operations: classification, embedding computation, regression, or clustering;
- LLM methods: semantic parsing, summarization, and textual reconstruction;
- Visualization components for final chart generation.

Decoding the matrix revealed three semantically coherent workflows:

- visual 1: ETL → ML → LLM → VIS, suitable for KPIs and trend analysis;
- visual 2: ETL → ML → LLM → VIS, producing semantic cluster views;
- visual 3: ETL → LLM → VIS, producing narrative-style or categorical summaries.

The existence of all four operation classes in the final chromosome confirms that the GA learned the implicit structure of real-world analytical reporting pipelines, even though no explicit constraints, beyond validity and type matching, forced such ordering. To validate that the evolved pipelines correspond to meaningful outputs, three synthetic visuals were generated:

- The regression forecast chart (**Fig. 5**) represents time-series prediction, illustrating the ML segment of the pipeline.
- Semantic embedding clusters (**Fig. 6**) represent LLM with ML synergy for contextual analytics.
- The categorical summary chart (**Fig. 7**) represents ETL aggregation followed by basic visualization.

These visuals emulate typical report components found in automated dashboards: trend analysis, semantic grouping, and KPI summaries.

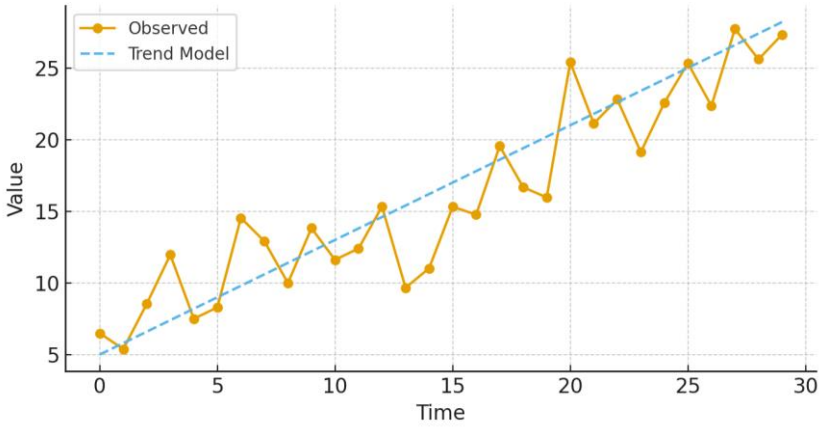


Fig. 5. Regression forecast chart.

The ability of the evolved pipelines to generate full visuals demonstrates that the GA was able to construct functional workflows, not just syntactic sequences of operations.

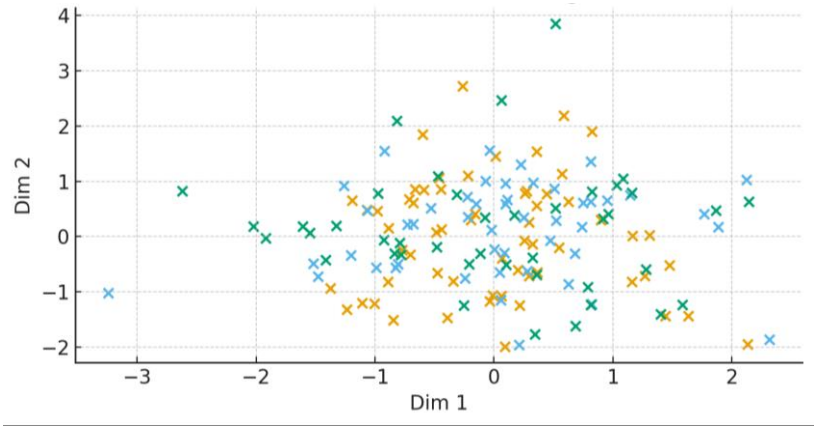


Fig. 6. Semantic embedding clusters.

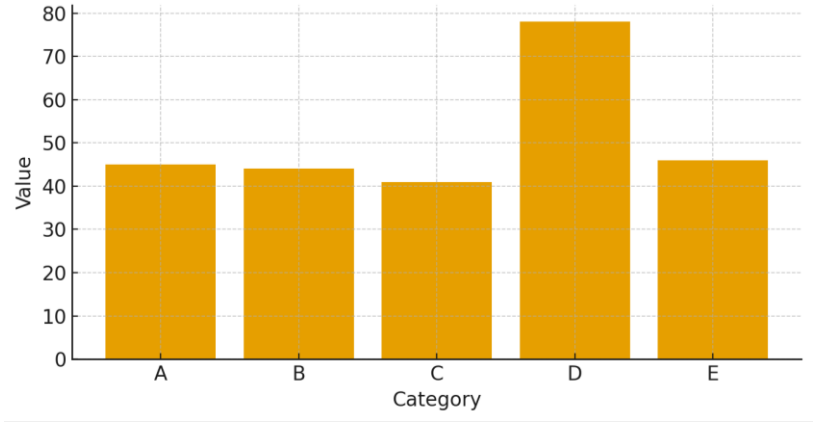


Fig. 7. Categorical summary chart.

The experimental results confirm several critical findings:

- Matrix chromosome encoding works. Encoding the chromosome as a matrix (pipeline-per-visual) supports parallel workflow evolution and allows GA to optimize complete multi-visual reports.
- GA can learn sustainable algorithms (pipelines). The convergence around 10-14 operations demonstrates that the fitness function effectively guides the algorithm toward sustainable ETL, ML, LLM, and VIS chains.
- Automatic multimodal report construction is feasible. The final pipelines successfully generated three distinct and meaningful visuals.
- Emergency structure matches human workflow design. Despite no explicit manual design, the GA consistently evolved a logical ordering: ETL first (ingestion, parsing, cleaning), ML/LLM next (interpretation, semantic enhancement) and visualization last (final chart generation).
- Population-level diversity was maintained. Although fitness converged, the distribution of pipeline structures remained diverse, indicating that mutation and crossover operators preserved exploration.

These results support the viability of using a tensor-based GA for industrial report automation, especially in heterogeneous data environments requiring simultaneous multimodal analytics.

CONCLUSION

This research demonstrates that evolutionary optimization, when combined with modern ML, LLMs, and agentic AI principles, provides an effective methodological foundation for automated report generation in heterogeneous industrial environments. The developed tensor-based GA model successfully evolves multi-stage, multi-visual analytic pipelines by representing each report as a matrix of parallel workflows and optimizing them jointly. Experimental results confirm that the proposed framework converges toward sustainable, interpretable, and structurally valid pipelines, balancing ETL, ML, LLM, and VIS operations into coherent analytic sequences.

The study also reveals that as the operational space expands, through the growth of meaning-space operations, improved code-generation capabilities, and the emergence of agentic AI systems, the effectiveness and scalability of evolutionary pipeline synthesis increase substantially. These technological developments amplify the adaptability and autonomy of the system, enabling dynamic reconfiguration of analytic workflows in response to new formats, data modalities, and reporting objectives. Consequently, evolutionary optimization becomes a foundational step for future agentic systems. It establishes an optimized, validated and semantically coherent action space from which AI agents can reliably plan, coordinate and execute reporting tasks. The findings thus position evolutionary algorithm-driven pipeline synthesis as a crucial enabling technology for next-generation autonomous analytic ecosystems.

The research introduces a novel encoding of analytic pipelines as a matrix L × N rather than a linear sequence, enabling simultaneous evolution of multiple report visuals within a single chromosome. This produces richer, more scalable, and structurally coherent solutions compared to classical GA pipeline representations.

The proposed framework integrates heterogeneous operation types into a single evolutionary unified multimodal operation space “ETL-ML-LLM-VIS”. This unification is unprecedented in existing AutoML or AutoETL systems. The experiments prove that complex reporting pipelines can emerge autonomously from evolutionary pressure, without manually encoded domain rules, which is an important step toward self-assembling analytics systems.

A key contribution of this work is the formalization of LLM-based semantic operations, such as summarization, multi-hop reasoning, entity and relation extraction, and contextual

normalization, as evolvable genetic components within the pipeline. Treating meaning-space transformations as genes places them on equal theoretical footing with classical feature engineering, ML, and ETL operations, thereby extending the domain of evolutionary computation from structural data manipulation into the realm of semantic cognition.

The study formulates a new design principle: before constructing agentic AI systems capable of autonomous report generation, the underlying operation space must be optimized through evolutionary search. This ensures that the action space available to agents is minimal, non-redundant, sustainable, and functionally validated.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author received no financial support for the research, writing and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The author declares that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

The author has read and agreed to the published version of the manuscript.

REFERENCES

[1] Fan, J., Han, F., & Liu, H. (2014). Challenges of Big Data analysis. National Science Review, 1(2), 293–314. https://doi.org/10.1093/nsr/nwt032

[2] Fernandes, A. A. A., Koehler, M., Konstantinou, N., et al. (2023). Data preparation: A technological perspective and review. SN Computer Science, 4(425). https://doi.org/10.1007/s42979-023-01828-8

[3] Ahlawat, P., Borgman, J., Eden, S., Huels, S., Iandiorio, J., Kumar, A., & Zakahi, P. (2023). A new architecture to manage data costs and complexity. BCG. https://on.bcg.com/3HOP7vQ

[4] Kwon, N., Comuzzi, M. (2023). Genetic Algorithms for AutoML in Process Predictive Monitoring. In: Montali, M., Senderovich, A., Weidlich, M. (eds) Process Mining Workshops. ICPM 2022. Lecture Notes in Business Information Processing, vol 468. Springer, Cham. https://doi.org/10.1007/978-3-031-27815-0_18

[5] Shi, K., Saad, S. (2023). Automated feature engineering for AutoML using genetic algorithms. In Proceedings of the 15th International Joint Conference on Computational Intelligence (pp. 450-459). SCITEPRESS. https://www.scitepress.org/Papers/2023/120904/120904.pdf

[6] Hernandez, J., Saini, A., Ghosh, A., Moore, J. (2025). The tree-based pipeline optimization tool: Tackling biomedical research problems with genetic programming and automated machine learning. Patterns. 6. 101314. 10.1016/j.patter.2025.101314. https://doi.org/10.1016/j.patter.2025.101314

[7] Jiao, J., Yuan, J. (2025). GA-PRE: A Genetic Algorithm-Based Automatic Data Preprocessing Algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '25). Association for Computing Machinery, New York, NY, USA, 1371–1378. https://doi.org/10.1145/3712256.3726312

[8] Polonskaia, I. S., Nikitin, N. O., Revin, I., Vychuzhanin, P., & Kalyuzhnaya, A. V. (2021, June). Multi-objective evolutionary design of composite data-driven models. In 2021 IEEE Congress on Evolutionary Computation (CEC) (pp. 926-933). IEEE. https://doi.org/10.1109/CEC45853.2021.9504773

[9] Jing, Z., Su, Y., Han, Y., Yuan, B., Liu, C., Xu, H., & Chen, K. (2024). When Large Language Models Meet Vector Databases: A Survey. arXiv. https://arxiv.org/abs/2402.01763

[10] Sequeda, J., Allemang, D., & Jacob, B. (2025). Knowledge graphs as a source of trust for LLM-powered enterprise question answering. Journal of Web Semantics, 85, 100858. https://doi.org/10.1016/j.websem.2024.100858
[11] Instaclustr. (2024). *Vector Databases and LLMs: Better Together*. https://www.instaclustr.com/education/open-source-ai/vector-databases-and-llms-better-together/
[12] DeepFA AI. (2025). Multi-Agent Systems in Artificial Intelligence. https://deepfa.ir/en/blog/multi-agent-systems-artificial-intelligence
[13] Ramachandran, A. (2025). *Revolutionizing Knowledge Graphs with Multi-Agent Systems: AI-Powered Construction, Enrichment, and Applications*. ResearchGate
[14] Mehta, V., Batra, N., Poonam, Goyal, S., Kaur, A., Dudekula, K. V., & Victor, G. J. (2024). Machine Learning Based Exploratory Data Analysis and Diagnosis of Chronic Kidney Disease (CKD). https://doi.org/10.4108/eetpht.10.5512
[15] Da Poian, V., Theiling, B., Clough, L., McKinney, B., Major, J., Chen, J., & Hörst, S. (2023). Exploratory data analysis (EDA) machine learning approaches for ocean world analogue-mass spectrometry. https://doi.org/10.3389/fspas.2023.1134141
[16] Nayak, U. (2025). AI-Powered Data Pipelines: Leveraging Machine Learning for ETL Optimization. Journal of Software Engineering and Simulation, 11(6), 134-136.
[17] Heffetz, Y., Vainstein, R., & Katz, G. (2019). DeepLine: AutoML Tool for Pipelines Generation using Deep Reinforcement Learning and Hierarchical Actions Filtering. arXiv. https://doi.org/10.48550/arXiv.1911.00061
[18] Chanda, D. (2024). Automated ETL Pipelines for Modern Data Warehousing: Architectures, Challenges, and Emerging Solutions. The Eastasouth Journal of Information System and Computer Science, 1(03), 209–212. https://doi.org/10.58812/esiscs.v1i03.523
[19] Lyashkevych, L., Lyashkevych, V., & Shuvar, R. (2024). Exploratory data analysis possibility in the meaning space using large language models. Electronics and Information Technologies, 1(25), 9, 102–116. http://dx.doi.org/10.30970/eli.25.9
[20] Chen, M., Tworek, J., Jun, H., et al. (2021). Evaluating large language models trained on code. arXiv:2107.03374. https://doi.org/10.48550/arXiv.2107.03374
[21] Wang, Y., Yin, W., Li, B., et al. (2023). CodeT5+: Open code large language models for code understanding and generation. arXiv:2305.07922. https://doi.org/10.48550/arXiv.2305.07922
[22] Rozière, B., Gehring, J., Gloeckle, F., et al. (2023). Code Llama: Open foundation models for code. arXiv:2308.12950. https://doi.org/10.48550/arXiv.2308.12950
[23] Zhang, Y., Wang, C., Xie, T., & Huang, J. (2023). A survey on program synthesis with large language models. arXiv:2311.07989. https://doi.org/10.48550/arXiv.2311.07989
[24] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. In Proceedings of the 37th International Conference on Neural Information Processing Systems (Article 2997, pp. 1–13). Curran Associates Inc. https://dl.acm.org/doi/10.5555/3666122.3669119.
[25] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023, October). Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th annual acm symposium on user interface software and technology (pp. 1-22). https://doi.org/10.48550/arXiv.2304.03442

СТАЛА ОПТИМІЗАЦІЯ АЛГОРИТМІВ ОБРОБКИ КОНСОЛІДОВАНИХ ДАНИХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ ТА ГЕНЕТИЧНИХ АЛГОРИТМІВ

Василь Ляшкевич
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005, Львів, Україна

АНОТАЦІЯ

Вступ. Автоматизація побудови аналітичних звітів у промислових компаніях набуває стратегічного значення через різноманіття форматів документів, збільшення обсягів даних та зростання вимог до швидкого формування багатокомпонентних аналітичних матеріалів. Традиційні ETL-конвеєри не справляються зі складністю сучасних інформаційних потоків, особливо коли у процес інтегруються машинне навчання, LLM-моделі та агентні системи. У зв'язку зі швидким прогресом генерації коду та автономних агентів, здатних виконувати складні аналітичні процедури, задача автоматичного конструювання звітних конвеєрів стає все більш перспективною та науково обґрунтованою.

Матеріали та методи. Запропоновано еволюційну модель побудови алгоритмів для опрацювання консолідованих звітів на основі генетичних алгоритмів (ГА). Для генерації звітів, алгоритм визначає конвеєр для побудови візуального компоненту. Популяція визначається як тензор, що забезпечує паралельну еволюцію множини незалежних робочих потоків. Операції класифіковано у чотири групи: ETL, ML, LLM та VIS. Функція пристосованості оцінює сталу довжину конвеєру, покриття ключових типів операцій та їх структурну узгодженість.

Результати. Експериментальні результати показали, що ГА швидко еволюціонує від випадкових NOP-домінованих структур до сталих, логічно узгоджених та функціональних конвеєрів довжиною 10-14 операцій. Найкращі хромосоми сформували три повноцінні візуальні компоненти: прогнозну регресійну модель, семантичну кластеризацію представлень вбудовування та категоріальну діаграму. Така еволюційна закономірність підтверджує, що комбіновані конвеєри можуть будуватися автоматично й адаптивно, а збільшення складності операцій у просторі "смислу", що є векторним простором вбудовування, разом із розвитком генерації коду та агентних архітектур.

Висновки. Запропонована модель демонструє ефективний механізм автоматизованого синтезу звітів з багатьма візуальними компонентами на основі еволюційних конвеєрів. Перспективність методу зростає із розвитком агентних систем ШІ та збільшенням кількості операцій у просторі смислу, що відкриває шлях до повністю автономних систем аналітичної звітності нового покоління.

Ключові слова: консолідовані дані, сталий оптимізаційний підхід, машинне навчання, генетичні алгоритми, LLM, аналітика даних

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
19 November, 2025	14 December, 2025	14 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 55–66
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.9:519.26(075.8)

CONCEPT OF ADAPTIVE SELECTION OF STRUCTURED LEARNING ALGORITHMS OF BAYESIAN NETWORKS BASED ON THEIR CHARACTERISTICS

Mariia Voronenko,
Odesa Polytechnic National University,
Shevchenko Avenue, 1, Odesa, 65044, Ukraine

Voronenko, M.O. (2025). Concept of Adaptive Selection of Structured Learning Algorithms of Bayesian Networks Based on Their Characteristics. *Electronics and Information Technologies*, 32, 55–66. https://doi.org/10.30970/eli.32.4

ABSTRACT

Background. Modern intelligent systems require efficient mechanisms for analysis, prediction, and decision-making. Bayesian networks allow for the efficient representation of causal relationships between variables. The quality and reliability of the constructed network directly correlate with the effectiveness of the final intelligent system.

Materials and Methods. Structural training of a Bayesian network involves determining the structure of a directed acyclic graph in which variables are related to each other. The quality of the structure has a decisive impact on the ability of the model to accurately represent conditional probabilities and on the efficiency of the training algorithms and the reliability of the model. The main problem limiting the structural learning of Bayesian networks is the computational complexity of the model. This fundamental complexity means that for multidimensional problems, it is impossible to perform a complete search of all possible structures and find a global optimum. This forces reliance on heuristic search methods and approximation algorithms and creates a constant need to balance the quality of structure determination and computational resources.

Results and Discussion. The formalized concept of adaptive selection of algorithms for structural learning is based on a systematic analysis of algorithm characteristics and data properties, which allows you to choose the most suitable algorithm for a particular case, optimize the trade-off between model quality and computational resources, and increase the generalizability of the model in practical scenarios.

Conclusion. The proposed concept of adaptive selection of algorithms for structural learning is a timely contribution to the field of stochastic dependence modeling. It successfully translates the process of selecting the optimal algorithm from a routine, heuristic, brute force method to a systematic, multivariate analysis. Its full implementation has the potential to significantly increase the reliability, accuracy, and computational efficiency of building Bayesian models in complex analytical domains.

Keywords: adaptive selection concept, Bayesian network, structured learning algorithm, model accuracy

INTRODUCTION

Modern intelligent systems are increasingly used for analysis, forecasting, and decision-making in complex multifactorial environments. In conditions of incompleteness, uncertainty, and the presence of noisy data, Bayesian networks (BNs) play a special role, which provides the ability to model cause-and-effect relationships between variables and

© 2025 Mariia Voronenko. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

make logical inferences. Bayesian networks are successfully used in such areas as medical diagnostics, data analysis in bioinformatics, financial forecasts, technical diagnostics, decision support systems, etc.

A key stage in building a BN is structural learning, which includes determining the graph of dependencies between variables, i.e., which variables in the network are related and in which direction the connections go. The quality of this structure has a decisive impact on the model's ability to accurately represent conditional probabilities, on the efficiency of inference algorithms, and on the ability to generalize. There are different approaches to structural learning: constraint-based, score-based, and hybrid methods, each of which has its own advantages and limitations depending on the data size, the noise level, the presence of missing values, or expert knowledge. For example, constraint-based methods analyze conditional independence between variables, while score-based approaches search for a structure that maximizes a certain value function [1]. Another important problem is computational complexity, which is the problem of structural learning that is NP-hard even with restrictions on the parent's number for each variable [1].

In this regard, there is a need to formalize the concept of algorithms adaptive selection for BN's structural learning, which would be based on a systematic analysis of the algorithm's characteristics (for example, time complexity, search space, sensitivity to noise, type of variables, missing data, expert estimates limitations) and on the data properties (number of variables, sample size, priori knowledge availability, etc.). Such an approach will allow choosing the most appropriate algorithm for a specific case, optimizing the trade-off between model quality and computational resources, and increasing the model's generalizability in practical scenarios.

Therefore, the purpose of this work is to develop a concept of adaptive algorithm selection for structural learning of Bayesian networks based on their characteristics, formalizing the criteria and parameters that determine the algorithm correspondence to the specific conditions of the problem, as well as empirical verification of the proposed approach.

The scientific novelty lies in the fact that:

1. the structured approach to the classification and structural learning algorithms comparison by a set of characteristics is proposed;
2. the set of adaptive selection criteria is defined, which considers not only the model quality, but also resources (time, memory), noise level, type of variables, and missing data;
3. the solution to the adaptive selection problem of a structural learning algorithm is proposed using the example case.

The implementation of the adaptive selection concept will contribute to increasing the efficiency of building Bayesian networks in various domains with complex data conditions, reducing time and computational costs, as well as increasing the reliability and accuracy of models.

REVIEW OF LITERATURE

Today, Bayesian models have become widespread and are comprehensively used in various industries. In [2], the authors carried out monitoring and diagnostics of a multi-stage production process using Bayesian methods. To perform medical diagnostics and differential diagnosis based on fuzzy and partially correct statistics, the authors [3] designed a static Bayesian network model. The authors [4-6] used probabilistic methods to model the supplier selection procedure based on the application of stability parameters. Dünder E., Cengiz M.A., and Koç H. investigated the impact of constraint-based algorithms on the Bayesian network structure quality in hybrid algorithms for medical research [7], and Ziegler V. investigated approximation algorithms for constrained Bayesian network structures [8]. There is also a number of studies devoted to learning the structure of Bayesian networks [9-13].

The main advantage of Bayesian networks is their robustness to incomplete, inaccurate, and noisy data [14,15]. In such complex cases, they are able to determine the most probable

outcome of events [16,17]. The study of the robustness of algorithms to noise (which is critically important), the ability to work with data with missing values, and the ability to take into account expert knowledge or a priori conditions, which are often present in practical applications, is devoted to the work [18]. Despite the large arsenal of existing methods, the question of adaptive selection of a structure learning algorithm for a specific task remains open. Many works compare the performance characteristics of methods: accuracy of structure recovery, time costs, data dimensionality, and others. For example, Scanagatta et al. (2018) analyze approximation algorithms for structure learning for large BNs, comparing the training time and quality of structure reproduction [19]. In the study [20], different score functions and heuristics are compared, and the noise influence is also studied.

PROBLEM STATEMENT

For a set of events $X^{(i)}, i = 1, \dots, N$ that are related, and a set of learning data $D = (d_1, \dots, d_n), d_i = \left\{x_i^{(1)} x_i^{(2)} \dots x_i^{(N)}\right\}$, is given. Here, the subscript is the observation amount, and the upper one is the variable amount, n–is the number of observations, each observation consists of $N(N \geq 2)$ variables, and each j -th variable $(j = 1, \dots, N)$ has $A^{(j)} = \{0,1, \dots, \alpha^{(j)} - 1\} \quad (\alpha^{(j)} \geq 2)$ conditions.

Based on a given training sample, you need to build an acyclic graph connecting the event sets $X_i,\ i = 1, \dots, N$. In addition, each BN structure $g \in G$ is represented by a set N of predecessors $\left(P^{(1)}, \dots, P^{(N)}\right)$, that is, for each vertex, $j = 1, \dots, N,\ P^{(j)}$ it is a variety of parent vertices, such that $P^{(j)} \subseteq \{X^{(1)}, \dots, X^{(N)}\} \backslash \{X^{(j)}\}$. We have events $X^{(i)}, i = 1, \dots, N$ that are affected by the uncertainties of a different nature. And also, we have data describing these events.

MATERIALS AND METHODS

The BN structure is learned using search algorithms at the 4th stage of the Bayesian network design sequence (**Fig. 1a**). Most of the existing methods for building the BN structure can be conditionally divided into two categories:

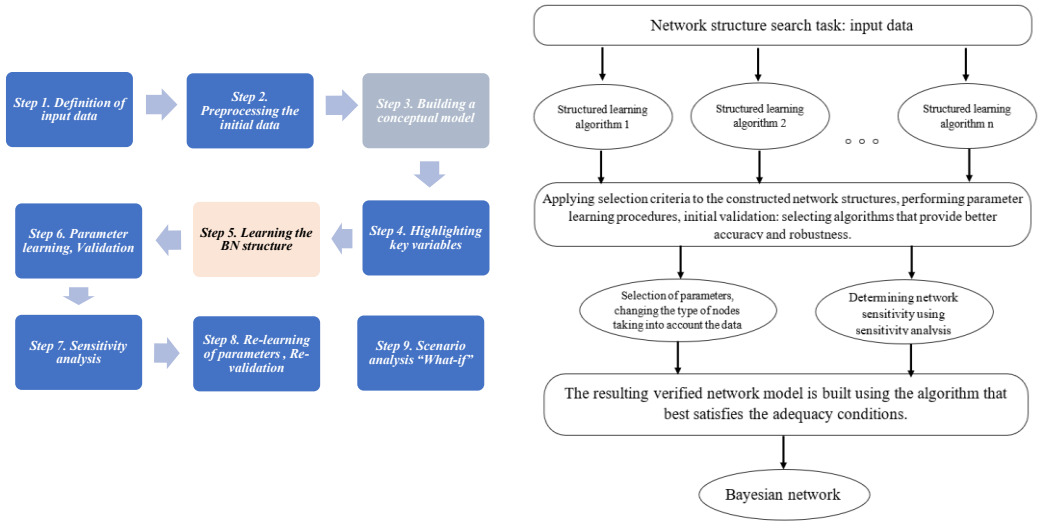
- methods based on evaluation functions (search & scoring);
- methods based on the application of the conditional independence test (dependency analysis) [21,22].

The adaptive selection concept is based on the ensemble method of selecting a structural learning algorithm that best fits the available data set (**Fig. 1b**). The ensemble method in structural learning of BN is an approach that, based on the results of several structural learning algorithms or several networks learned on the same data, allows obtaining a more accurate final structure of the network model than that built using a single algorithm.

The main idea is that different algorithms can fall into different local optima or find structures that are close in estimate, especially on limited, incomplete, or noisy data. The choice is made in favour of the model that, after repeated validation and sensitivity analysis, has the highest accuracy.

The advantages of the ensemble method are to reduce the influence of noise in the data and randomness in the selection of the initial structure. The final model demonstrates higher predictive accuracy. Due to the generation of structures by different algorithms, the ensemble is more likely to cover a wider space of structure search.

The main idea of the Path Condition (PC) algorithm is to obtain a set of conditionally independent and dependent nodes according to some statistical tests. PC checks the statistical tests for conditional independence for all variable pairs, except for the specified restrictions. An undirected link is added between each pair of variables in which conditional independence was not found.



a) b)

Fig. 1. a) Scheme of the BN's phased design, b) detailing the 5th stage "Learning the BN structure".

The undirected graph obtained in this way is called the skeleton of the learning structure [23]. Then, pairs of directed links are determined in such a way that they meet in a node, and that ensure the undirected cycles absence.

The next step is to find the directions of the links, which can be obtained from the conditional probability table. The remaining links will be arbitrarily directed, ensuring that no directed cycles will occur. In their work, Dempster and Druzdzel experimentally confirmed that the PC algorithm is quite robust to multivariate analysis. For comparison, they performed a graph structure search first using the PC algorithm and then using the Bayesian search approach [23]. Suppose we have a set of variables $X = (X_1, \ldots, X_n)$ with a global probability distribution over them P. We will denote by the letter A the subset of the variables X. By $I(A,B/C)$ we will denote that the sets A and B are conditionally independent of C. The PC algorithm assumes confidence probabilities.

This means that there exists a directed acyclic graph G such that the independence relations between the variables in X are exactly those represented in G by the d-partition criterion [23]. The PC algorithm assumes a procedure that can recognize when $I(A,B|C)$ is verified (tested) on a graph G. At first, it tries to find the skeleton which underlies the undirected graph, and in the last step, it does the edge orientation. The pseudocode of the algorithm PC can be represented as follows:

```
1. Start with a complete undirected graph G'
2. i = 0
3. Repeat
4.     For each X ∈ X
5.       For each Y ∈ ADJX
6.          Test whether ∃S ⊆ ADJX – {Y} with |S| = i and I (X, Y |S)
7.          If this set exists
8.             Make SXY = S
9.             Remove X – Y link from G'
10.         End If
11.      End For
12.   End For
13.   i = i + 1
14. Until |ADJX| ≤ i, ∀X.
```

The Greedy Thick Thinning (GTT or Greedy) structure learning algorithm is described by Cheng in [22]. GTT starts with an empty graph and repeatedly adds an edge without creating a cycle, maximizing the marginal likelihood P(D|S) until the addition of an edge leads to a positive increase (this is the “thickening” phase). It then repeatedly removes edges until the removal of an edge leads to a positive increase in P(D|S) (this is the “thinning” phase). The algorithm is quite efficient due to its susceptibility to the local maxima trap. This is a universal method for learning the structure of a graph. The probability tables are populated using expectation maximization [22]. The Greedy algorithm implements a “greedy” search for the existence of possible arcs from and the attribute variable A. If all attributes depend on the class variable, then the Bayesian network will have the structure of an extended simple Bayesian classifier. However, the purpose of Greedy is to determine which of these dependencies are really necessary. In addition, at each step, Greedy extends the revealed structure of dependencies between the class variable and the attribute variables by applying the Augmenter operator. It starts its work with an empty set of descendants and at each step adds a new descendant that is optimal in accordance with the value q of the network structure measure. The descendants of a node C are the elements of the set γ, and the set λ contains those attribute variables that do not directly depend on C. For each configuration of descendants γ, the dependencies between the attribute variables A are determined by an appropriately chosen Augmenter operator. The input information for the Augmenter operator is the dependencies between the class variable C and the attribute variables A, which are represented by the set $\gamma_{i-1} \cup \{X\}$. It learns additional dependencies between attribute variables using a set of training data D. Greedy selects the network that corresponds to the configuration of descendants λ with the largest value of the quality function q. The pseudocode of the GTT algorithm can be represented as follows:

Greedy $(D, q,$ Augmenter$)$

1. $\gamma \leftarrow \emptyset$ { a set of variables that depend on a class variable }
2. $\lambda \leftarrow \mathbf{A} = \{A_1, \ldots, A_{n-1}\}$ { a set of variables that do not depend on a class variable }
3. $\hat{q} \leftarrow -\infty$ { the highest value of the quality measure }
4. **for** $i = 1, \ldots, n-1$ **do**
5. Choose the attribute $A \in \lambda$, that maximizes the quality measure $q(B, D)$, for the network $B \leftarrow$Augmenter$(\gamma \cup \{A\}, \lambda \backslash \{A\}, D, q)$
6. **if** $q(B, D) > \hat{q}$ **then**
7. $\hat{B} \leftarrow B$
8. $\hat{q} \leftarrow q(B, D)$
9. $\gamma \leftarrow \gamma \cup \{A\}$
10. $\lambda \leftarrow \lambda \backslash \{A\}$
11. **return**

The Naive Bayes network structure learning algorithm is a structure learning method that is included in the category of structured learning algorithms only because it creates the structure and parameters of a Bayesian network directly from the data. The structure of a Naive Bayes network is not learned, but rather fixed by the assumption that the class variable is the sole parent of all remaining functional variables, and there are no other connections between the nodes of the network. The Naive Bayes structure assumes that the functions are independent of the class variable, which leads to inaccuracies. The pseudocode of the Naive Bayes algorithm can be represented as follows:

Class-Depend (S, γ, D)

1. $\hat{S} \leftarrow S$
2. **for** each variable $\gamma_i \in \gamma$ **do**

3. Add a directed arc $(C \to \gamma_i)$ to the network structure $\hat{S}$
4. **return** $\hat{S}$

The operation of the Naive Bayes Tree-Augmenter (TAN) algorithm is described by Friedman in [24,25]. The Tree Augmented structure learning algorithm starts with a Naive Bayes structure (i.e., one in which the class variable is the only parent of all remaining object variables) and adds a relationship between the feature variables to account for the possible dependency between them due to the class variable. The algorithm restricts only one additional parent of each feature variable (except for the class variable, which is the parent of each feature variable). Note that the Naive Bayes structure assumes independence of the feature from the class variable, which leads to inaccuracies when they are not independent. The TAN algorithm is simple and has been found to perform reliably better than Naive Bayes. The pseudocode of the Tree Augmented structure learning algorithm can be represented as follows:

Tree-Augmenter $(D, q,$ Augmenter$)$

1. $G \leftarrow \emptyset$
2. **for** each pair of variables $\{A_i, A_j\} \subset \gamma \cup \lambda$ such that $A_i \neq A_j$ **do** $w_{ij} \leftarrow I_\gamma(A_i A_j)$
3. add an undirected arc $(A_i - A_j)$ to the graph G
4. $T \leftarrow Maximum - Spanning - Tree(G, \boldsymbol{w})$
5. Order the arcs of an undirected tree T, by choosing one node as the root and setting the directions of all arcs from it. Then transform it into a Bayesian structure S.
6. $\hat{B} \leftarrow Class - Depend(S, \gamma, D)$
7. return $\hat{B}$

The Augmented Naive Bayes (ANB) structure learning algorithm is a semi-scientific structure learning method based on the Bayesian search approach. The ANB algorithm starts with a Naive Bayes framework (i.e., in which the class variable is the sole parent of all remaining feature variables) and adds a link between the feature variables to account for possible dependency between them due to the class variable. There is no limit to the number of additional connections included in each of the feature variables unless imposed by one of the algorithm's parameters. Note that the Naive Bayes framework assumes that the features are independent of the class variable, which leads to inaccuracies when they are not independent. The pseudocode of the ANB algorithm can be represented as follows:

ANB $(D, q,$ Augmenter$)$

1. $\gamma \leftarrow \emptyset$ { a set of variables that depend on a class variable }
2. $\lambda \leftarrow \boldsymbol{A} = \{A_1, \ldots, A_{n-1}\}$. {a set of variables that do not depend on a class variable }
3. $\hat{q} \leftarrow -\infty$ {the highest value of the quality measure }
4. **for** $i = 1, \ldots, n-1$ **do**
5. choose the attribute $A \in \lambda$, that maximizes the quality measure $q(B, D)$, for the network $B \leftarrow$Augmenter$(\gamma \cup \{A\}, \{A\}, \emptyset, D)$
6. **if** $q(B, D) > \hat{q}$ **then**
7. $\hat{B} \leftarrow B$
8. $\hat{q} \leftarrow q(B, D)$
9. $\gamma \leftarrow \gamma \cup \{A\}$
10. $\lambda \leftarrow \lambda \backslash \{A\}$
11. **return**

Fig. 2 presents the results of designing Bayesian network models using five structured learning algorithms.

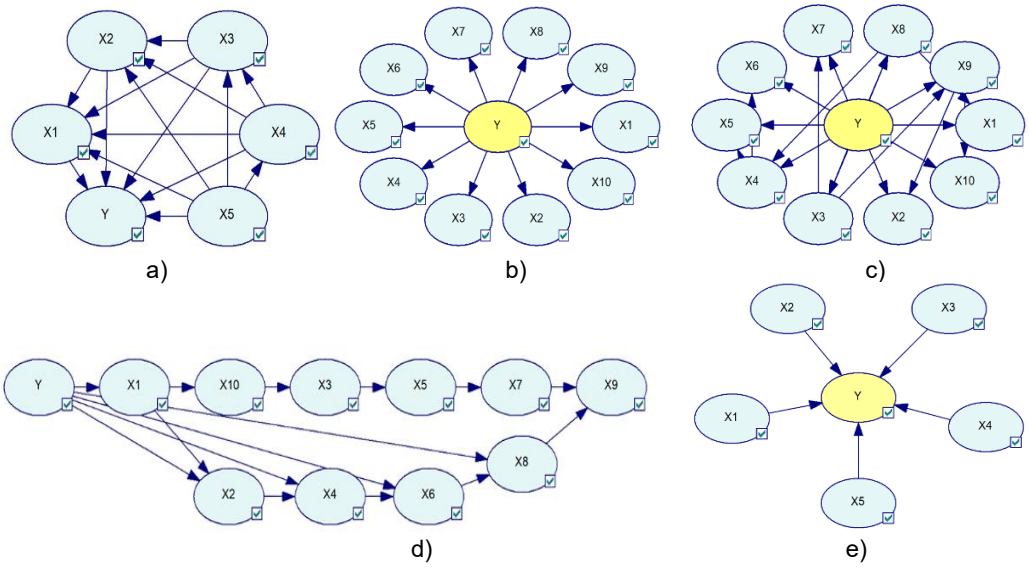


Fig. 2. Bayesian network models built using the algorithm: a) PC, b) ANB, c) TAN, d) GTT, e) NB.

RESULTS AND DISCUSSION

We will consider the implementation of the concept of choosing a structural learning algorithm that best fits the available data using the research example in [26]. This research concerns the modelling of investments in the transport sector and is best suited for experiments on selecting the best structure learning algorithm. Conceptual model of a Bayesian network for calculating the dependence of GDP growth of Ukraine on the volume of investments in the transport sector is shown in **Fig. 3**.

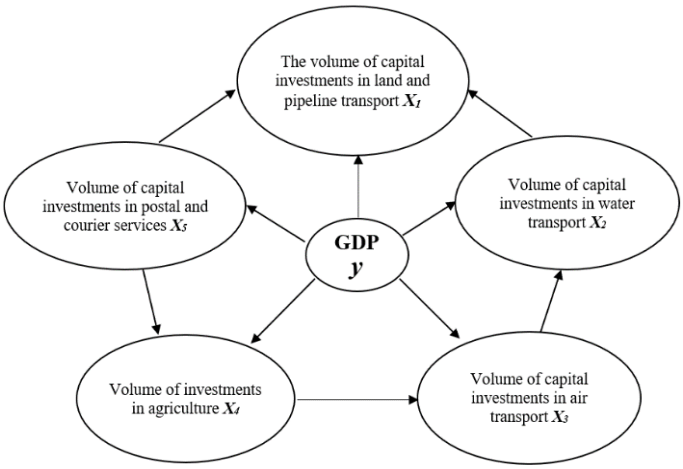


Fig. 3. Conceptual model of a Bayesian network for calculating the dependence of GDP growth of Ukraine on the volume of investments in the transport sector [26].

At the first stage of BN design in the GeNie2.3, we take the initial type of nodes to be General, each node has 5 states from s_1 to s_5. The following macroeconomic indicators for a period of 5 years were taken as experimental data to calculate the dependence of Ukraine's GDP growth on the volume of investments in the transport sector:

- x_1 – the volume of investments in land and pipeline transport in actual prices;
- x_2 – the volume of investments in water transport;
- x_3 – the volume of investments in air transport;
- x_4 – the volume of investments in warehousing and auxiliary activities in the transport sector;
- x_5 – the volume of investments in postal and courier activities.

The set of available data can be divided into two sets: 16 measurements is learning sample A, 8 measurements is test sample B [26].

At the first stage, we designed a structural model using the algorithms presented in the GeNie2.3. **Fig. 4** shows the first 3 algorithms that were applied to the available data: these are PC, TAN i ANB.

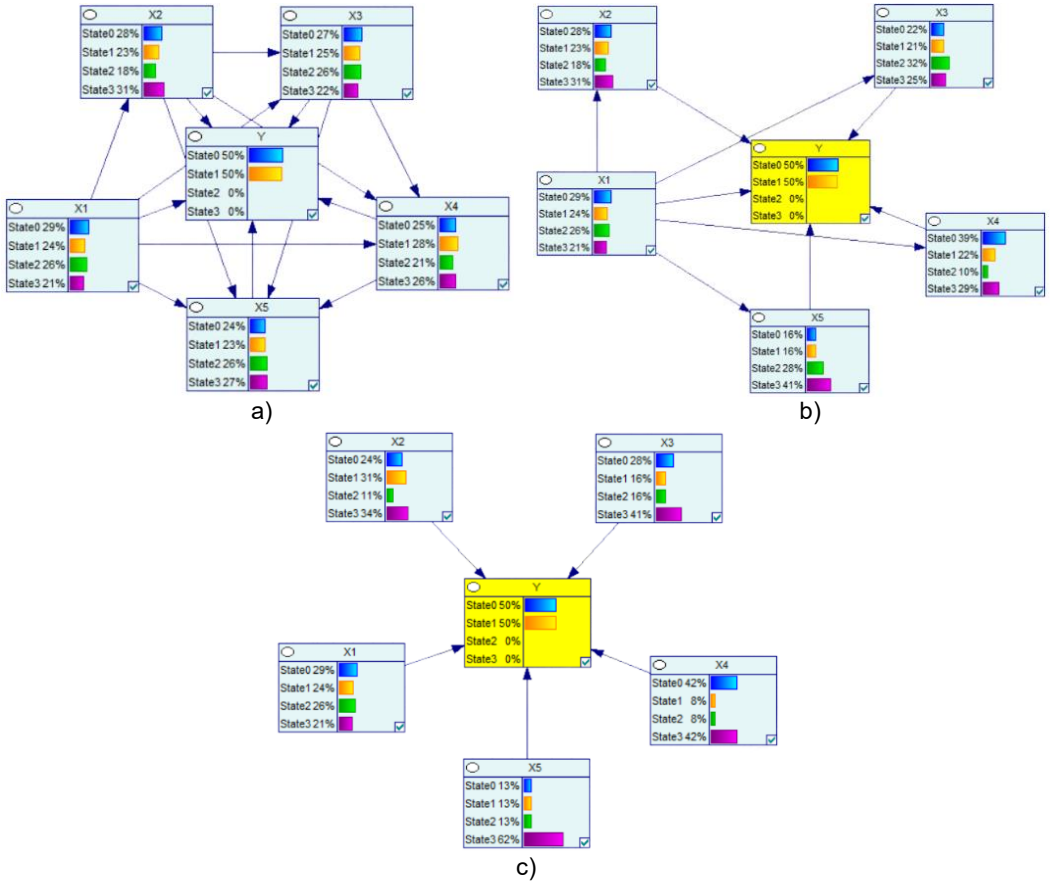


Fig. 4. Bayesian network models built using the algorithm: a) PC, b) TAN, c) ANB.

Similarly, structural models were built using the Bayesian search and Greedy algorithms. Then, the primary parameterization and validation of each network were sequentially carried out. As a result of the experiment on the structural learning method selection, five Bayesian networks were obtained, each of which consists of 6 nodes. After parametric learning, the primary validation of each network was carried out. Among the designed networks, using different structural learning algorithms, we select the network that meets the requirements for accuracy and adequacy of the model. The Greedy algorithm turned out to be an adequate method when working with the existing data set. The selection results are presented in the **Fig. 5**.

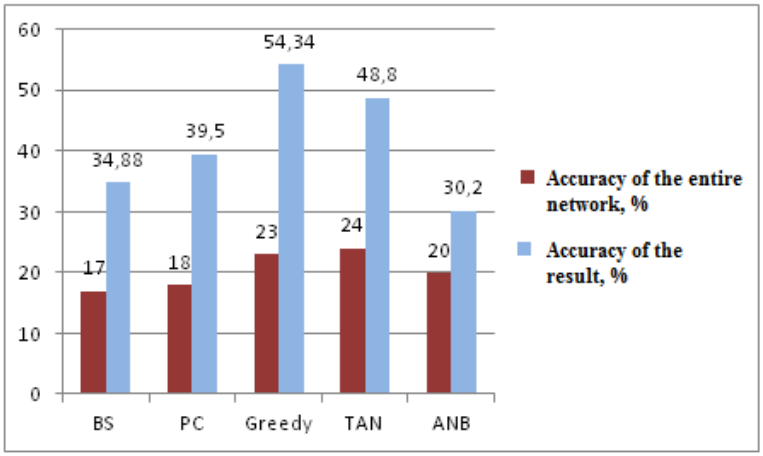


Fig. 5. Selection of the structural learning method [26].

Fig. 5 shows the results of the validation of five Bayesian network models built using five structured learning algorithms. We see that the three algorithms TAN, Greedy, and PC have the highest accuracy, so we will continue to work with them, and we discard the others. At the next stage, the type of all nodes was changed to Noisy-MAX with four states $s_1 \dots s_4$. This may slightly increase the computational complexity, but such a replacement helps to improve accuracy. The fact is that when the model is small and has a small number of nodes and arcs, the increase in computational complexity will be minimal, so in the case of small models, replacing the GENERAL node type with the Noisy-MAX node type is quite justified.

We have three network models, the most adequate, built using the TAN, Greedy, and PC structural learning algorithms; the data file remains unchanged. We conduct re-learning of parameters, re-validation, and sensitivity analysis. The accuracy comparison is shown in the **Fig. 6**.

The accuracy of the result in this case corresponds to the accuracy of forecasting the category of growth of the country's gross domestic product, depending on the volume of investments in the transport sector.

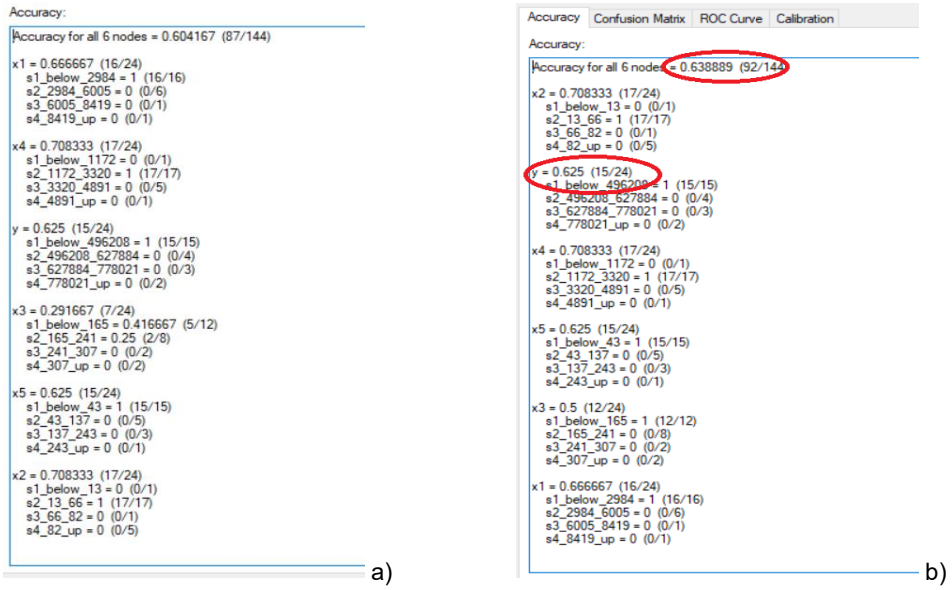


Fig. 6. Overall accuracy of the network and accuracy of the result after: a) primary and b) revalidation.

CONCLUSION

For the investment modelling problem with a critically small sample size (N = 16), the Greedy algorithm, as a score-based heuristic, turned out to be the most adequate, which indicates its advantage in combating overtraining compared to algorithms that rely on statistical power, such as PC. Further introduction of strong prior assumptions at the parametric level using Noisy-MAX nodes led to an increase in the overall accuracy of the network from 60.42% to 63.89%, confirming that the parametric knowledge inclusion is an important criterion for an adaptive approach in the data absence.

The proposed concept of adaptive selection of Bayesian network structural learning algorithms is a timely contribution to the field of stochastic dependence modeling. It successfully translates the process of selecting the optimal algorithm from routine, heuristic, brute force to systematic, multivariate analysis. Its full implementation has the potential to significantly improve the reliability, accuracy, and computational efficiency of BN construction in complex analytical domains.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author received no financial support for the research, authorship, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The author declares that she has no competing interests.

AUTHOR CONTRIBUTIONS

The author has read and agreed to the published version of the manuscript.

REFERENCES

[1] Kitson, N.K., Constantinou, A.C., Guo, Z. et al. A survey of Bayesian Network structure learning. Artif Intell Rev 56, 8721–8814 (2023). https://doi.org/10.1007/s10462-022-10351-w.

[2] Wolbrecht E., D'Ambrosio B., Paasch R., Kirby D. Monitoring and diagnosis of a multistage manufacturing process using Bayesian networks. Ai Edam. 1998;14:53–67. https://doi.org/10.1017/S0890060400141058.

[3] Nikovski D. Constructing Bayesian networks for medical diagnosis from vague and partially correct statistics. IEEE Trans. Knowl. Data Eng. 2000;12:509–516. https://doi.org/10.1109/69.868904.

[4] Hosseini S., Barker K. A Bayesian network model for resilience-based supplier selection. Int. J. Prod. Econ. 2016; 180: 68–87. https://doi.org/10.1016/j.ijpe.2016.07.007.

[5] Hosseini S., Barker K. Modeling infrastructure resilience using Bayesian networks: A case study of inland waterway ports. Comput. Ind. Eng. 2016; 93: 252–266. https://doi.org/10.1016/j.cie.2016.01.007.

[6] Hosseini S., Barker K. A Bayesian network model for resilience-based supplier selection. Int. J. Prod. Econ. 2016; 180: 68–87. https://doi.org/10.1016/j.ijpe.2016.07.007.

[7] Dünder E., Cengiz M.A., Koç H. Investigation of the impacts of constraint-based algorithms to the quality of Bayesian network structure in hybrid algorithms for medical studies. J. Adv. Sci. Res. 2014; 5: 8–12. https://sciensage.info/index.php/JASR/article/view/175.

[8] Ziegler V. Approximation algorithms for restricted Bayesian network structures. Inf. Process. Lett. 2008;108:60–63. https://doi.org/10.1016/j.ipl.2008.03.015.

[9] Li S., Zhang J., Sun B., Lei J. An incremental structure learning approach for Bayesian network; Proceedings of the 26th Chinese Control and Decision Conference; Changsha, China. 31 May–2 June 2014; pp. 4817–4822. https://doi.org/10.1109/CCDC.2014.6853036
[10] Chickering D.M., Dan G., Heckerman D. Learning Bayesian networks is NP-hard. Networks. 1994; 112: 121–130.
[11] Xu J.-G., Zhao Y., Chen J., Han C. A structure learning algorithm for Bayesian network using prior knowledge. J. Comput. Sci. Technol. 2015;30:713–724. https://doi.org/10.1007/s11390-015-1556-8.
[12] Amirkhani H., Rahmati M., Lucas P., Hommersom A. Exploiting experts' knowledge for structure learning of Bayesian networks. IEEE Trans. Pattern Anal. Mach. Intell. 2016; 39: 2154–2170. https://doi.org/10.1109/TPAMI.2016.2636828.
[13] Zhang Y., Zhang W., Xie Y. Improved heuristic equivalent search algorithm based on maximal information coefficient for bayesian network structure learning. Neurocomputing. 2013; 117: 186–195. https://doi.org/10.1016/j.neucom.2013.02.015.
[14] Voronenko M., Mikhailova E., Savina N., Lurie I., and Lytvynenko V., "A Bayesian Network Approach to Measure the Impact of Factors Characterizing the State of Educational Institutions on the Gross Domestic Product", in Proc. 13th International Conference on Advanced Computer Information Technologies, ACIT 2023, pp. 316–320, 2023. https://doi.org/10.1109/ACIT58437.2023.10275641.
[15] Voronenko M., Zhunissova U., Smailova S., Lytvynenko L., Savina N., Mulesa P., and Lytvynenko V., "Using Bayesian Methods in the Task of Modeling the Patients' Pharmacoresistance Development", IAPGOŚ, 2, pp.77-82, 2022, http://doi.org/10.35784/iapgos.2968.
[16] Romanko O., Voronenko M., Savina N., Zhorova I., Wójcik W. and Lytvynenko V., "The Use of Static Bayesian Networks for Situational Modeling of National Economy Competitiveness", in Proc. IEEE International Conference on Advanced Trends in Information Theory (ATIT), pp. 501-505, Kyiv, Ukraine, 2019, https://doi.org/10.1109/ATIT49449.2019.9030515.
[17] Voronenko M., Lurie I., Boskin O., Zhunissova U., Baranenko R. and Lytvynenko V., "Using Bayesian Methods for Predicting the Development of Children Autism", in Proc. IEEE International Conference on Advanced Trends in Information Theory (ATIT), pp. 525-529, Kyiv, Ukraine, 2019, https://doi.org/10.1109/ATIT49449.2019.9030523.
[18] Li H, Guo H. A Hybrid Structure Learning Algorithm for Bayesian Network Using Experts' Knowledge. Entropy (Basel). 2018 Aug 20; 20 (8): 620. https://doi.org/10.3390/e20080620. PMID: 33265709; PMCID: PMC7513154
[19] Scanagatta, M., Corani, G., de Campos, C. P., & Zaffalon, M. (2018). Approximate structure learning for large Bayesian networks. Machine Learning, 107(8-10), 1209-1227. https://doi.org/10.1007/s10994-018-5701-9.
[20] Stefano Beretta, Mauro Castelli, Ivo Goncalves, Roberto Henriques, Daniele Ramazzotti, "Learning the structure of Bayesian Networks: A quantitative assessment of the effect of different algorithmic schemes", 2018, https://doi.org/10.48550/arXiv.1704.08676.
[21] Cheng J., Bell D. A. Liu W. Learning belief networks from data: an information theory based approach // Proceedings of the sixth international conference on information and knowledge management (CIKM 1997), Las Vegas (Nevada), November 10-14. 1997. P. 325-331. https://doi.org/10.1145/266714.266920.
[22] Cheng J., Greiner R., Kelly J., Bell D. A., Liu W. Learning Bayesian networks from data: an information-theory based approach. The artificial intelligence journal (AIJ). 2002. Vol. 137. P. 43-90.
[23] Dempster A. P., Laird N. M., Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society: series B (methodological). 1977. Vol. 39, № 1. P. 1-22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

[24] Freedman D., Kisilev P. Fast mean shift by compact density representation // Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2009. P. 1818-1825. https://doi.org/10.1109/CVPR.2009.5206716.
[25] Friedman N., Murphy K., Russell S. Learning The Structure Of Dynamic Probabilistic Networks // Proc. Fourteenth Conference On Uncertainty In Artificial Intelligence. 1998. https://doi.org/10.48550/arXiv.1301.7374.
[26] Lytvynenko V., Savina N., Krejci J., Voronenko M., Yakobchuk M., and Kryvoruchko O. Bayesian Networks' Development Based on Noisy-MAX Nodes for Modeling Investment Processes in Transport. In Proc. 2-nd International Workshop «Modern Machine Learning Technologies and Data Science» (MoMLeT&DS-2019), pp. 1-10, Shatsk, Ukraine, 2019, http://ceur-ws.org/Vol-2386/.

КОНЦЕПЦІЯ АДАПТИВНОГО ВИБОРУ АЛГОРИТМІВ СТРУКТУРНОГО НАВЧАННЯ БАЙЄСІВСЬКИХ МЕРЕЖ НА ОСНОВІ ЇХНІХ ХАРАКТЕРИСТИК

Марія Вороненко
Національний університет «Одеська політехніка»,
проспект Шевченка, 1, м. Одеса, 65044, Україна

АНОТАЦІЯ

Вступ. Сучасні інтелектуальні системи потребують ефективних механізмів аналізу, прогнозування та прийняття рішень. Байєсівські мережі дозволяють ефективно представляти причинно-наслідкові зв'язки між змінними.

Матеріали та методи. Структурне навчання байєсівської мережі включає визначення структури орієнтованого ациклічного графа, в якому змінні пов'язані між собою. Якість структури має вирішальний вплив на здатність моделі точно представляти умовні ймовірності і на ефективність алгоритмів навчання та надійність моделі. Основною проблемою є обчислювальна складність моделі. Це означає, що для багатовимірних задач неможливо виконати повний пошук усіх можливих структур, і знайти глобальний оптимум. Це змушує покладатися на евристичні методи пошуку та алгоритми апроксимації і створює постійну потребу в балансуванні якості визначення структури та обчислювальних ресурсів.

Результати. Концепція адаптивного вибору алгоритмів для структурного навчання базується на систематичному аналізі характеристик алгоритму та на властивостях даних, що дозволяє вибрати найбільш підходящий алгоритм для конкретного випадку, оптимізувати компроміс між якістю моделі та обчислювальними ресурсами, підвищити узагальнюваність моделі в практичних сценаріях.

Висновки. Запропонована концепція адаптивного вибору алгоритмів структурного навчання переводить процес вибору оптимального алгоритму з рутинного, евристичного методу грубої сили на систематичний, багатовимірний аналіз. Її повна реалізація має потенціал для значного підвищення надійності, точності та обчислювальної ефективності побудови Байєсівських моделей у складних аналітичних областях.

Ключові слова: концепція адаптивного вибору, Байєсівська мережа, алгоритм структурного навчання, точність моделі.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
06 November, 2025	05 December, 2025	08 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 67–86
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.932.2

COMPREHENSIVE SPATIAL-GEOMETRIC EVALUATION OF KEYPOINT DETECTORS

Andriy Fesiuk**, Yuriy Furgala
Faculty of Electronics and Computer Technologies
Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine

Fesiuk A. V., Furgala Y. M. (2025). Comprehensive Spatial-Geometric Evaluation of Keypoint Detectors. *Electronics and Information Technologies*, 32, 67–86. https://doi.org/10.30970/eli.32.5

ABSTRACT

Background. Local features are essential components of modern computer vision systems, such as SLAM and 3D reconstruction. Traditional evaluation protocols for keypoint detection mainly focus on geometric accuracy and repeatability, often neglecting the spatial structure of the point distribution. This complicates algorithm selection for applications where uniform image coverage and the absence of excessive local clustering are important. This work aims to conduct a comprehensive comparison of keypoint detectors using an extended set of metrics that account for both geometric accuracy and the spatial properties of features.

Materials and Methods. The study was conducted on the HPatches dataset using six detectors: SIFT, SURF, ORB, BRISK, KAZE, and AKAZE. Keypoint filtering and geometric verification of correspondences were performed using USAC. Matching quality was assessed through the geometric metrics MMA, Repeatability, and Verification Ratio. Spatial analysis used the metrics CUI, RI, and SCS. To compare keypoint detection methods, a quality index Q was introduced that integrates geometric and spatial indicators.

Results and Discussion. The study showed that selecting points by response strength significantly improves matching accuracy for SIFT, ORB, and BRISK, but may lead to local redundancy of keypoints. KAZE and AKAZE demonstrated the best overall balance, achieving high accuracy along with more uniform scene coverage. ORB tended to form dense clusters in high-contrast regions, thereby reducing its structural effectiveness, whereas SURF consistently delivered high performance regardless of the keypoint selection strategy.

Conclusion. The proposed evaluation method allows a consistent analysis of the geometric and spatial properties of keypoint detectors. It shows that, for a fixed number of keypoints, the performance of the final method depends not only on the geometric accuracy of matches but also on the features of the spatial point distribution. It was observed that the keypoint selection process, especially response-based selection, systematically affects both geometric and spatial characteristics. The Q quality index combines these aspects into a single metric. It can be used to compare detection methods in scenarios that require both reliable matches and well-balanced scene coverage.

Keywords: feature detection, spatial distribution, geometric metrics, image matching.

INTRODUCTION

Computer vision systems are widely used in navigation, 3D scene reconstruction, visual tracking, and augmented reality [1-5]. In many of these applications, keypoints and their descriptors play a central role in establishing correspondences between images. The

© 2025 Andriy Fesiuk & Yuriy Furgala. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

quality of matching directly affects the accuracy of geometric model estimation, tracking stability, and the reliability of subsequent processing stages. Therefore, objective comparison of keypoint detectors remains relevant despite the large body of research in this area.

Traditionally, detector performance is evaluated using repeatability, matching accuracy, the inlier ratio after geometric verification, and the homography estimation error. These measures characterize the stability of feature localization and the geometric correctness of matches. At the same time, the spatial properties of keypoint sets are often considered only to a limited extent: non-uniform image coverage, local clustering, or the presence of “blind zones” can reduce the robustness of geometric estimates even when the classical metrics remain acceptable. Some studies introduce coverage or uniformity indicators; however, they are typically analyzed in isolation and not consistently linked to geometric validation outcomes [6, 7].

An additional practical factor is the keypoint selection strategy when the number of keypoints is limited. Selecting the “strongest” features by detector response can improve matching accuracy, but it may also change the spatial profile of the keypoint set and its structural consistency with the scene. A coherent set of metrics that simultaneously accounts for geometric correctness and spatial-structural properties is therefore essential for a well-grounded comparison of detection methods.

In this study, we propose a comprehensive evaluation scheme for keypoint detectors that combines traditional geometric measures, namely MMA, Repeatability, Verification Ratio, with spatial-structural metrics: Coverage Uniformity Index (CUI), Redundancy Index (RI), and Scene Consistency Score (SCS). For an overall comparison, we introduce a quality index Q that integrates geometric and spatial characteristics. Experiments are performed on the HPatches dataset [8, 9] using detectors such as SIFT [10], SURF [11], KAZE [12], AKAZE [13], ORB [14], and BRISK [15], with geometric correspondence validation utilizing USAC [16].

MATERIALS AND METHODS

The experimental study was conducted on the HPatches dataset [8, 9]. This dataset contains images of planar scenes with varying levels of geometric distortion, for which ground-truth homography matrices are available, enabling precise verification of correspondences. HPatches provides two main types of sequences: viewpoint and illumination. In this work, we analyze the viewpoint sequences. Twelve sequences were selected: apprentices, azzola, busstop, cartooncity, dirtywall, london, posters, samples, sunseason, tabletop, talent, and vitro, which contain enough keypoints for all considered methods. For each image sequence, the original image and five images of the same scene with progressively increasing viewpoint changes were used. The ground-truth transformation between the reference image and each target image is a 3×3 projective homography, which was used for overlap computation and geometric evaluation. For each sequence, "reference image - current viewpoint" pairs were formed. Examples of image sequences are presented in **Fig. 1a**, and the first images of all used sequences are shown in **Fig. 1b**.

The study covers six algorithms for local feature detection and description: the floating-point methods SIFT, SURF, and KAZE, and the binary methods ORB, BRISK, and AKAZE. For each detector, the performance dependence on the number of keypoints is examined, with the number of keypoints ranging from 500 to 4000 in increments of 500. To evaluate how feature selection influences subsequent matching quality, two selection strategies were compared:

- raw-order: Selection of the first N points in the order returned by the detection method.

Fig. 1. Examples from the HPatches dataset: (a) the 'vitro' image sequence; (b) all scenes used in the experiments.

- top-response: Sorting points by detector response strength, then selecting the top N features with the highest response. This simulates a scenario where, under a limited keypoint budget, priority is given to the strongest local structures. The response for each method was calculated as follows [17]:
  - SIFT: The amplitude of the extremum in the Difference of Gaussians (DoG) pyramid, which correlates with contrast.
  - SURF: The determinant of the Hessian matrix used for detecting "blob-like" structures.
  - ORB and BRISK: Metrics based on the Harris corner detector for ORB and on AGAST for BRISK.
  - KAZE and AKAZE: The determinant of the Hessian matrix calculated in a nonlinear scale space, which better preserves object boundaries than traditional Gaussian blurring.

Matching was performed using the Brute-Force method, searching for the two nearest neighbors in descriptor space. Euclidean distance was used for SIFT, SURF, and KAZE descriptors, while Hamming distance was applied for ORB, BRISK, and AKAZE. Preliminary filtering involved Lowe's ratio test with a threshold of 0.75 [10].

Geometric verification was conducted using the USAC_DEFAULT method with a homography model and a 3-pixel threshold [16, 18]. Based on the USAC results, an inlier mask was extracted, allowing the separation of points consistent with the global projection. In this experiment, spatial metrics were computed for all detected points and inliers after filtering, enabling comparison of detector properties before and after geometric verification of correspondences.

The Coverage Uniformity Index quantitatively assessed the uniformity with which points covered the image plane. The image plane was divided into a regular grid of $M = 8 \times 8 = 64$ rectangular cells. For each set of points, the number of keypoints N_i in the i cell was counted, and a normalized distribution was formed.

$$p_i = \frac{N_i}{N}, \tag{1}$$

where N is the total number of points.

The obtained empirical distribution is compared with a perfectly uniform distribution, representing a scenario where each cell contains an equal proportion of keypoints:

$$u_i = \frac{1}{M} \tag{2}$$

The total variation distance measures the deviation between p_i and u_i, and the CUI is defined as:

$$CUI = 1 - \frac{1}{2}\sum_{i=1}^{M} |p_i - u_i|, \quad CUI \in [0,1]. \tag{3}$$

CUI values close to 1 correspond to almost uniform image coverage, while low values indicate an intense concentration of keypoints in a few cells. Conceptually, this metric is based on the coverage evaluation method proposed in [6], in which the spatial distribution of keypoints complements repeatability in assessing detector quality.

The Redundancy Index was used to measure the local redundancy of keypoints. The need to control redundancy is driven by the fact that traditional detector quality metrics can be biased toward methods that produce spatially overlapping or overly dense detections [19, 20]. Such behavior can result in misleadingly high precision values. Additionally, recent research on redundancy removal techniques has demonstrated that explicitly considering local density and distances between points helps reduce descriptor duplication and improves overall image registration efficiency [20, 21], supporting the goal of the proposed evaluation.

The RI calculation was performed for the set of points $P = (x_i, y_i)_{i-1}^{N}$ on the image of size $W \times H$. To eliminate the dependence on the resolution, the coordinates of the points were normalized to the unit square $[0,1]^2$:

$$\bar{x} = \frac{x_i}{W}, \quad \bar{y} = \frac{y_i}{H}. \tag{4}$$

Next, for each point $(\bar{x}_i, \bar{y}_i)$, the number of neighbors within a normalized radius *r* was counted. The radius is defined as a fraction of the unit square diagonal:

$$r = p\sqrt{2}, \tag{5}$$

where $p = 0.02$, and the factor $\sqrt{2}$ represents the diagonal of the unit square $[0.1]^2$, giving an interpretable definition of the local neighborhood. The choice of the *p* coefficient is based on its representing the typical size of a descriptor's support region, which helps identify points with strongly correlated vector descriptions due to significant spatial overlap.

Local redundancy was evaluated with the saturation function:

$$c_i = \min\left(\frac{n_i - 1}{K_{ref}}, 1\right), \tag{6}$$

where $K_{ref} = 15$, reflecting the empirical limit of information saturation: a concentration of more than 15 points within a single patch does not enhance the geometric model

estimation, for which 4-8 points are enough, but it only adds unnecessary computational load during the matching process.

The global RI score was calculated by averaging the local values:

$$RI = \frac{1}{N}\sum_{i=1}^{N} c_i\,, \quad RI \in [0,1]. \tag{7}$$

High RI values indicate strong local clustering, while low values suggest sparse and unique point arrangements.

In addition to evaluating spatial uniformity, an essential aspect of the analysis is the detector's ability to adapt to the scene's semantic content, as shown in **Fig. 2**. Traditional methods often exhibit a bias toward certain feature types, such as corners, while neglecting other informative elements in the image; this can lead to data loss when reconstructing scenes with complex geometry. To examine this criterion, an automatic image segmentation procedure was developed to categorize the image into three structural classes: high-contrast corner/texture regions T, contour/edge regions C, and homogeneous regions F. Reference masks are generated from gradient analysis [22-24]. The corner mask T is generated using the Harris detector, followed by Gaussian blur and thresholding at the 97.5th percentile of response. The Canny detector produces the edge mask C and excludes pixels already included in mask T. The mask F consists of pixels with low Sobel gradient magnitude = below the 25th percentile that do not intersect with T and C. This method ensures the mutual exclusivity of these sets, enabling the unique classification of each image region.

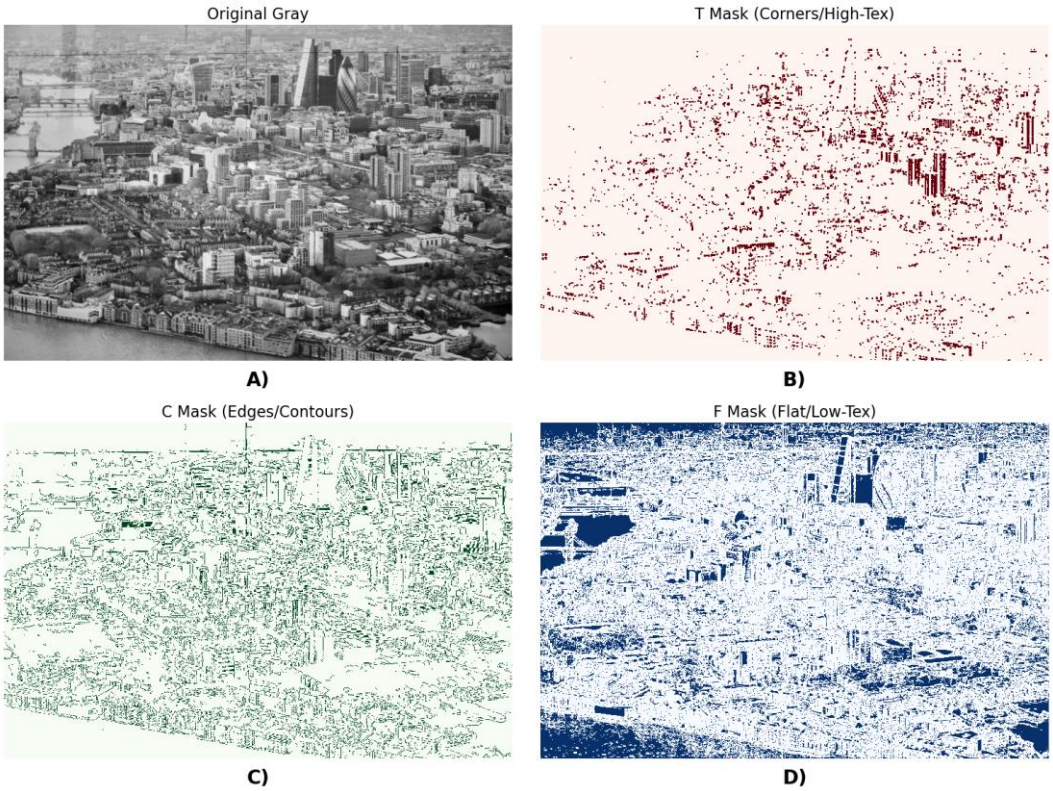


Fig. 2. Structural masks for scene analysis: (A) original image; (B) texture/corner mask T; (C) contour/edge mask C; (D) homogeneous (flat) mask F.

Based on the obtained masks, the relative areas were calculated:

$$\alpha_T = \frac{|T|}{HW}, \quad \alpha_C = \frac{|C|}{HW}, \quad \alpha_F = \frac{|F|}{HW}, \tag{8}$$

where H and W represent the image size, and $|\cdot|$ is the number of pixels in the corresponding mask. Then, the areas were normalized within the structural part of the image:

$$A_{str} = \alpha_T + \alpha_C + \alpha_F, \quad \alpha_T = \frac{\alpha_T}{A_{str}}, \quad \alpha_C = \frac{\alpha_C}{A_{str}}, \quad \alpha_F = \frac{\alpha_F}{A_{str}}. \tag{9}$$

For a given set of keypoints, the number of points falling into each mask (n_T, n_C, n_F) was counted, and then the proportions were calculated:

$$n_{str} = n_T + n_C + n_F, \quad \beta_T = \frac{n_T}{n_{str}}, \quad \beta_C = \frac{n_C}{n_{str}}, \quad \beta_F = \frac{n_F}{n_{str}}. \tag{10}$$

The consistency score of the keypoint distribution with the scene structure was defined as the complement of the total variation distance between α and β [7]:

$$SCS = 1 - \frac{1}{2}(|\beta_T - a_T| + |\beta_C - a_C| + |\beta_F - a_F|), \quad SCS \in [0, 1]. \tag{11}$$

Scene Consistency Score values close to 1 indicate that keypoints appropriately cover structural types in proportion to their actual presence in the scene. In contrast, lower values indicate a systematic bias, such as an excessive concentration on contours or a dominance of detections in texture or corner regions.

To evaluate classical performance metrics, standard protocols used in the HPatches benchmark [8, 9] were applied.

Repeatability was only calculated within the common overlap region of the images. A point x in the first image was considered repeatable if, after being projected through homography $H_{1\to2}$ into the second image, there was a point $\acute{x}$ within $\varepsilon = 3$ pixels of the projected position $n(H_{1\to2}\acute{x})$. To prevent multiple counts of the duplicate detection, "one-to-one" matching was employed [25, 26].

Mean Matching Accuracy was calculated on the set of points that passed Lowe's ratio test [10]. The MMA value was defined as the ratio of correct matches, with a reprojection error $\tau = 3$ in pixels, relative to the Ground Truth homography among all detected pairs [8].

To evaluate the efficiency of keypoint use, the Verification Ratio was employed. Unlike MMA, it is not based on the true homography, but on the results of filtering using the USAC method.

$$VR = \frac{N_{inliers}}{N_{total}}, \tag{12}$$

where $N_{inliers}$ is the number of points consistent with the found geometric model, N_{total} is the total number of detected points.

For a comprehensive comparison of detectors based on geometric and spatial-structural quality, an integral Quality Index (Q) was formulated. Its construction is based on

a weighted linear combination of two aggregated components: the geometric component *G* and the structural-spatial component *S*.

The geometric component *G* characterizes the accuracy of correspondence establishment. It is defined as the arithmetic mean of the normalized values of repeatability, matching accuracy, and verification ratio obtained after the filtering stage:

$$G = \frac{1}{3}(MMA_{\tau=3} + Rep_{\varepsilon=3} + VR). \tag{13}$$

The structural-spatial component S evaluates scene coverage quality and the detector's adaptation to scene content by combining metrics such as structural consistency, distribution uniformity, and local redundancy:

$$S = \frac{1}{3}\left(CUI_{filt} + (1 - RI_{filt})^2 + SCS_{filt}\right), \tag{14}$$

where $(1 - RI_{filt})^2$ indicates a preference for low local redundancy on a $[0, 1]^2$ scale, and its quadratic form emphasizes differences between sets with moderate and high local redundancy while also diminishing the influence of small RI values in the low redundancy range.

The weighting coefficients were assigned according to the golden ratio principle [27]. Since geometric accuracy is essential for most computer vision tasks, it was assigned a higher weight of 0.62, whereas structural-spatial quality is an additional important factor with a weight of 0.38.

$$Q = 0.62 \cdot G + 0.38 \cdot S. \tag{15}$$

The validation of the quality index is performed by analyzing the correlation between the *Q* values and the first principal component *PC1*, obtained via *PCA* [28] for all metrics, which allowed verifying the consistency of the proposed integral indicator with the multidimensional data structure.

The standard HPatches protocol is used for Repeatability, MMA, and Verification Ratio [8,9]. CUI adopts the coverage concept from [6] in a grid-based form. RI, SCS, and the Quality Index are introduced in this study. Masks are computed using classical Harris, Sobel, and Canny operators [22-24], whereas T/C/F partitioning and normalization are defined in this work.

RESULTS AND DISCUSSION

The analysis of the Coverage Uniformity Index in **Fig. 3** shows that, for all detectors, coverage uniformity improves as the quantity of keypoint *N* increases. After USAC geometric validation, CUI_{filt} values are consistently lower than CUI_{raw} because inliers form a more selective subset of correspondences. For KAZE and AKAZE, top_response provides noticeably more uniform coverage even at low keypoint counts; as *N* increases, both modes converge, reaching high CUI_{raw} values of about 0.65-0.70 at *N* = 4000. After filtering, these two methods maintain some of the best CUI_{filt} values among the detectors, around 0.40-0.45 for large *N*.

For SIFT and BRISK, selecting the strategy has the most significant impact: switching to top_response significantly increases CUI, even for small *N*. For SIFT at *N* = 500, CUI_{raw} the value rises from 0.16 to 0.47, and this improvement persists across the entire keypoint range. After USAC filtering, CUI values decrease, but the difference between methods

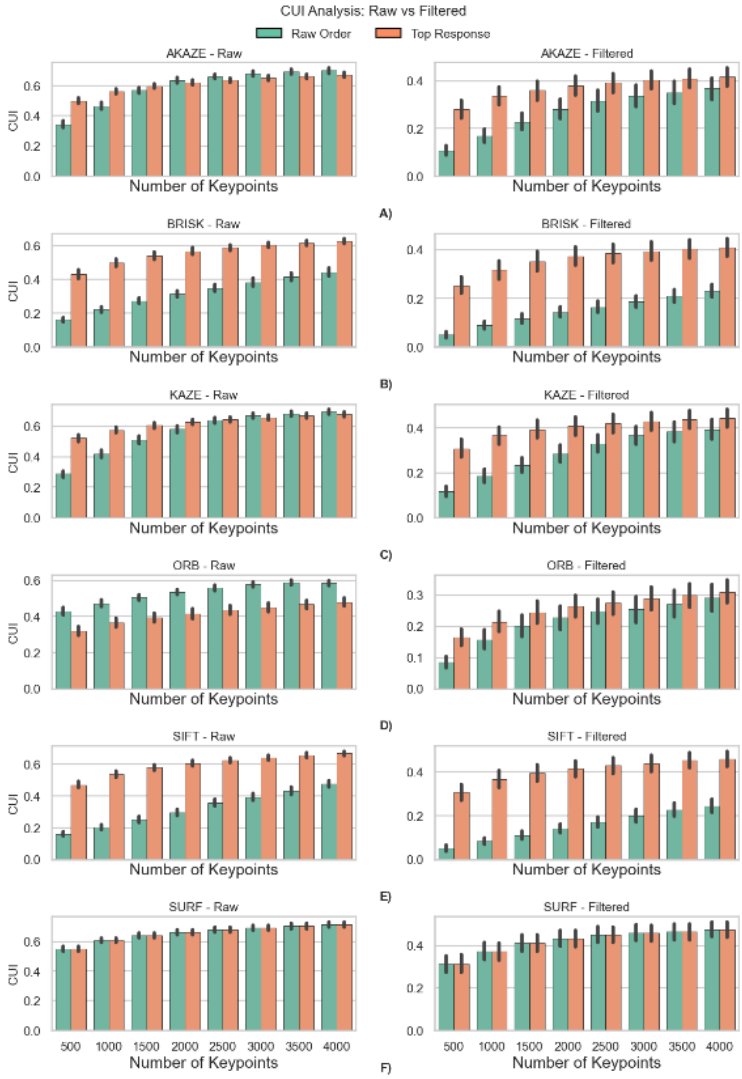


Fig. 3. Average CUI vs. the number of keypoints for raw sets (left) and filtered (right) across detectors under two selection strategies: A) AKAZE; B) BRISK; C) KAZE; D) ORB; E) SIFT; F) SURF.

remains steady, showing a more uniform spatial distribution of inliers with the top_response approach.

A different pattern appears for ORB: on unfiltered points, the raw_order mode yields a higher CUI, but after filtering, the difference between modes diminishes and shifts in favor of top_response. SURF shows the most consistent behavior: CUI is independent of the selection mode, indicating that the detector returns values already sorted by top_response, with CUI_{raw} values staying high at around 0.6 or above for all *N*. After filtering, SURF also maintains some of the highest CUI_{filt} values, approximately 0.47 at *N* = 4000. In conclusion, SURF offers the best average uniform coverage, followed by KAZE and AKAZE. SIFT and BRISK rank next, for which the top_response mode is essential for achieving high CUI.

The results of the Redundancy Index metric in **Fig. 4** show that as the number of keypoints increases, local redundancy tends to grow for most detectors in the unfiltered set RI_{raw}. This indicates a tendency for points to densify within the "strongest" local structures.

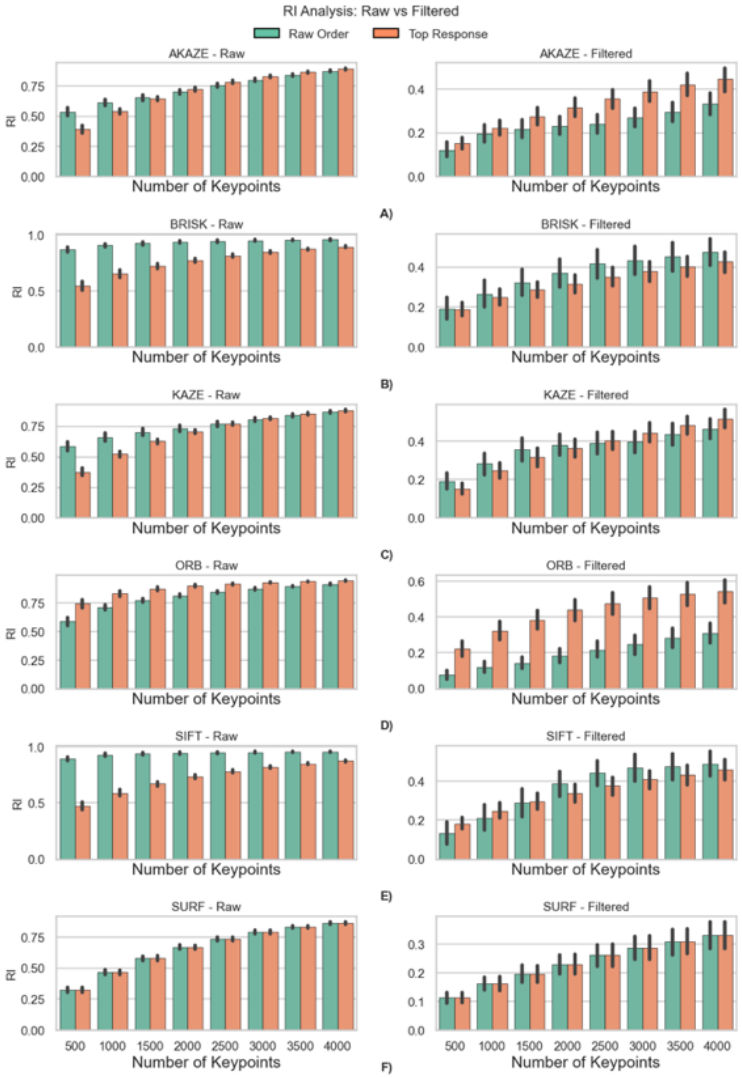


Fig. 4. Average RI vs. the number of keypoints for raw sets (left) and filtered (right) across detectors under two selection strategies: A) AKAZE; B) BRISK; C) KAZE; D) ORB; E) SIFT; F) SURF.

The highest RI_{raw} values are observed for SIFT and BRISK in raw_order mode, averaging approximately 0.94 and 0.93 across the entire range, which suggests pronounced clustering. Switching to top_response for these methods significantly changes RI values, reducing redundancy to around 0.72 for SIFT and 0.77 for BRISK. This indicates that selecting keypoints based on response produces a “sparser” point distribution.

For AKAZE, KAZE, and SURF, the unfiltered RI_{raw} values are lower, generally ranging from 0.65 to 0.75. The effect of top_response is weaker and depends on the method: for KAZE, changes are minimal, while for AKAZE, selecting by response slightly decreases redundancy. In contrast, for ORB, the opposite trend occurs: top_response increases local clustering even at the RI_{raw} level, averaging about 0.89 versus 0.80 for raw_order, which is consistent with ORB's tendency to concentrate points in a limited set of corner or high-contrast regions.

After filtering, RI_{filt} decreases significantly and shifts to a range characteristic of moderate inlier density, mainly between 0.2 and 0.4, indicating the selection of more

structurally consistent and spatially "cleaned" matches. AKAZE and SURF show the lowest inlier values, approximately 0.24 for raw_order and 0.24 for SURF in both modes. For AKAZE, top_response increases RI_{filt} by about 0.32, indicating that inliers remain relatively sparse. Conversely, for ORB, the top_response mode substantially increases inlier redundancy: RI_{filt} rises from approximately 0.20 to 0.43, indicating denser local clusters even after geometric verification. Overall, the RI results suggest that the keypoint selection strategy can significantly alter the local structure of the set: for SIFT and BRISK, top_response reduces redundancy, whereas for ORB it enhances clustering, particularly at the inlier level.

Fig. 5 shows the dependencies of the Scene Consistency Score. For the selected HPatches scenes, the average area proportions of structural zones $a_T = 0.083$, $a_T = 0.278$, and $a_F = 0.639$, indicating that low-texture regions are predominant on average. Under these conditions, SURF exhibits the highest consistency in the inlier distribution

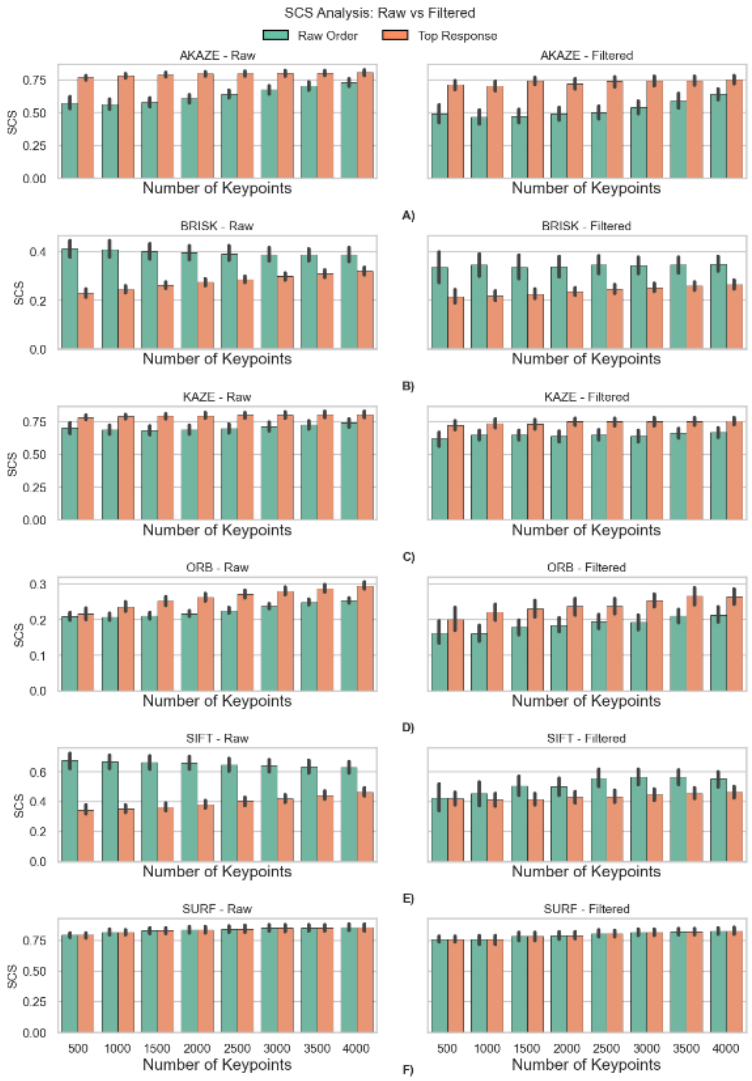


Fig. 5. Average SCS vs. the number of keypoints for raw sets (left) and filtered (right) across detectors under two selection strategies: A) AKAZE; B) BRISK; C) KAZE; D) ORB; E) SIFT; F) SURF.

aligned with the scene structure: SCS values remain high, at approximately 0.83 before filtering and 0.79 after filtering. High values are also typical of KAZE and AKAZE, especially in the top_response mode, where SCS_{filt} stays around 0.73-0.74.

In contrast, ORB and BRISK have the lowest SCS values, averaging 0.3 and 0.4 before filtering, and 0.2-0.25 after USAC processing, indicating a significant discrepancy between scene structure and the actual inlier distribution. For SIFT, a pronounced dependence on the selection strategy is observed: in raw_order, SCS_{raw} it remains relatively high at approximately 0.65, whereas top_response reduces consistency to approximately 0.4, and the advantage of raw_order persists after filtering.

The interpretation of these differences is supported by the structural analysis in **Fig. 6**, where the proportions β_T, β_C, β_F are compared before and after filtering and for the two keypoint selection strategies. For ORB and BRISK, inliers are sharply skewed toward corner/texture zones: β_T reaches approximately 0.8-0.9, while the contribution of

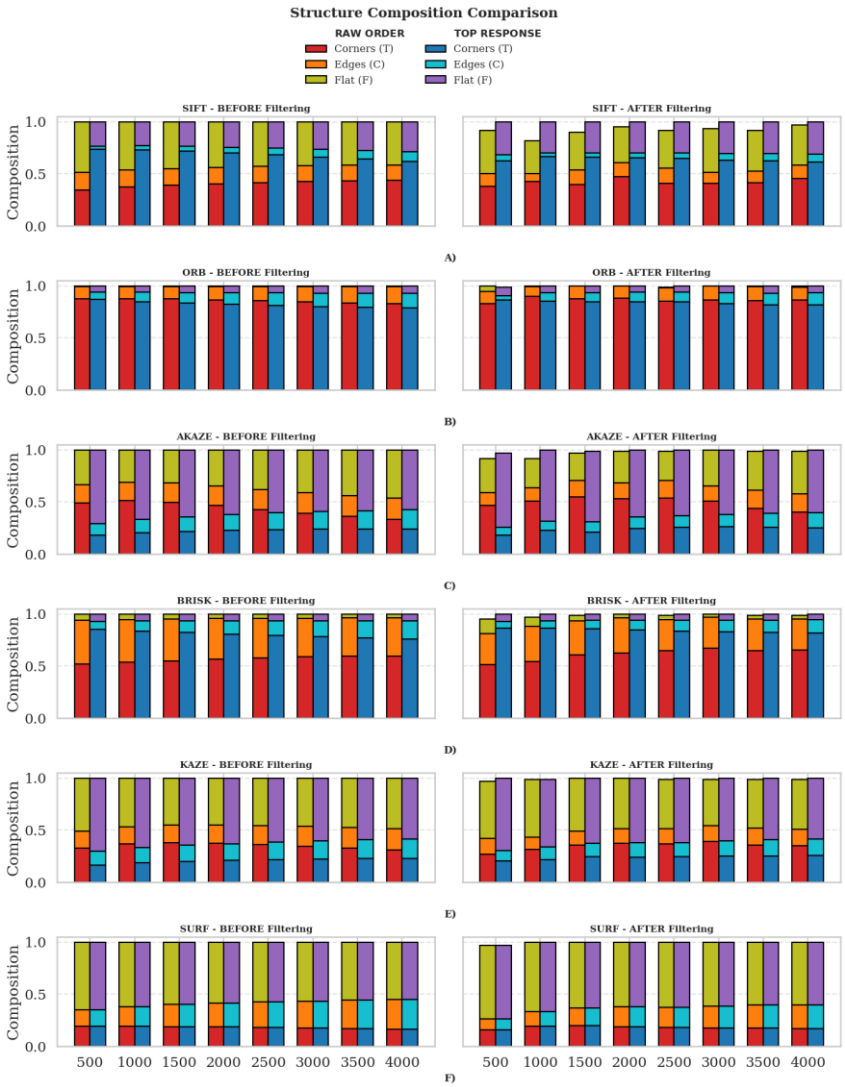


Fig. 6. Average Structure Composition vs. the number of keypoints for raw sets (left) and filtered (right) across detectors under two selection strategies: A) SIFT; B) ORB; C) AKAZE; D) BRISK; E) KAZE; F) SURF.

homogeneous regions β_F is minimal; this effect is more potent in top_response, and for ORB, it persists even after USAC processing. For SIFT, the top_response mode also causes a shift toward corner structures, which is consistent with the decline in SCS values. In contrast, KAZE, AKAZE, and SURF produce values with a significant proportion of low-texture regions β_F and moderate contributions from β_T and β_C, with their proportions changing only slightly after filtering. It is precisely this stability and structural "neutrality" that corresponds to the high SCS values observed across a wide range of keypoint counts.

Fig. 7 presents the MMA values, which indicate the percentage of matches that agree with the Ground Truth homography within a 3-pixel threshold. Two main patterns are observed across most detectors: in raw_order mode, MMA values increase with the number of keypoints, and switching to top_response consistently yields higher MMA values at lower keypoint counts than raw_order.

SIFT and ORB show the most substantial reliance on top_response. Even at *N* = 500, MMA scores increase from 0.27 to 0.75 for SIFT and from 0.41 to 0.66 for ORB, demonstrating that selecting by response significantly boosts the number of geometrically correct matches at low keypoint counts. When averaged across all *N*, this is reflected in an increase in MMA from approximately 0.55 to 0.75 for SIFT and from 0.54 to 0.71 for ORB. For AKAZE, KAZE, and BRISK, this effect is also present but less marked. Their mean values climb from roughly 0.64 to 0.76 for AKAZE, from 0.71 to 0.76 for KAZE, and from 0.72 to 0.80 for BRISK. The MMA scores for SURF stay around the 0.7 range.

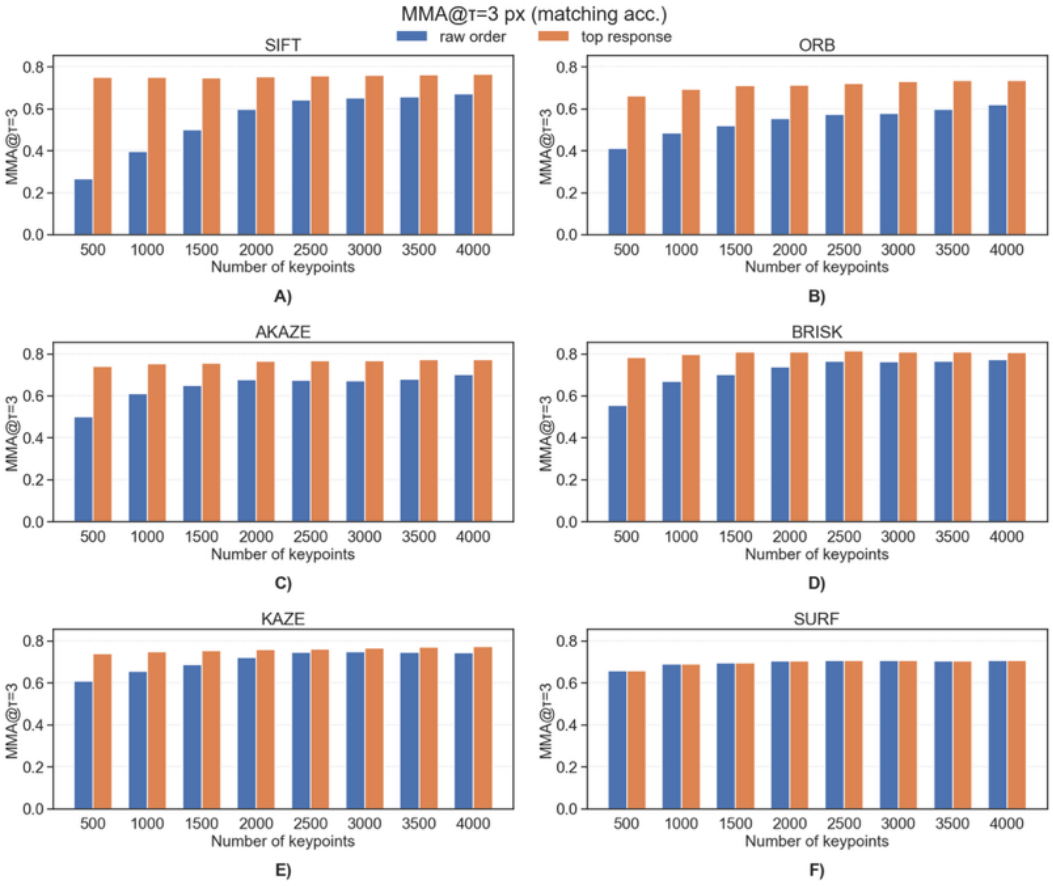


Fig. 7. Average MMA vs. the number of keypoints across detectors under two selection strategies: A) SIFT; B) ORB; C) AKAZE; D) BRISK; E) KAZE; F) SURF.

Overall, the MMA results confirm that selecting keypoints by response enhances the geometric correctness of matches across all methods, with SIFT and ORB showing the greatest sensitivity to the selection strategy.

Fig. 8 shows the Repeatability values, which measure the proportion of keypoints with a matching detection in the paired image within a 3-pixel threshold in the overlap region. Across all detectors, repeatability increases with the number of keypoints; most methods show the most significant improvements at low *N* values, then gradually level off.

KAZE exhibits the most consistent behavior, maintaining high repeatability levels around 0.55-0.60 across nearly all values of *N*, with minimal dependence on the point selection method. This suggests strong keypoint repeatability in viewpoint change tasks. BRISK and AKAZE achieve similar performance levels, although the effect of the selection strategy is more pronounced for BRISK. Switching to top_response consistently boosts repeatability, particularly at lower keypoint counts, reaching about 0.58-0.62 at larger *N*. SURF shows moderate yet highly stable repeatability, roughly in the range of 0.48-0.52.

In raw order mode, SIFT exhibits lower Repeatability, with values ranging from 0.25 to 0.38 across the entire set of keypoints. Nevertheless, selection based on the top response consistently improves the metric, raising it to approximately 0.42-0.50. Regarding ORB, the influence of top response is most significant at low keypoint counts, with a noticeable increase between *N*=500 and *N*=1500. Conversely, at higher keypoint counts, the disparity between the modes diminishes substantially, with both curves converging to approximately 0.57-0.59.

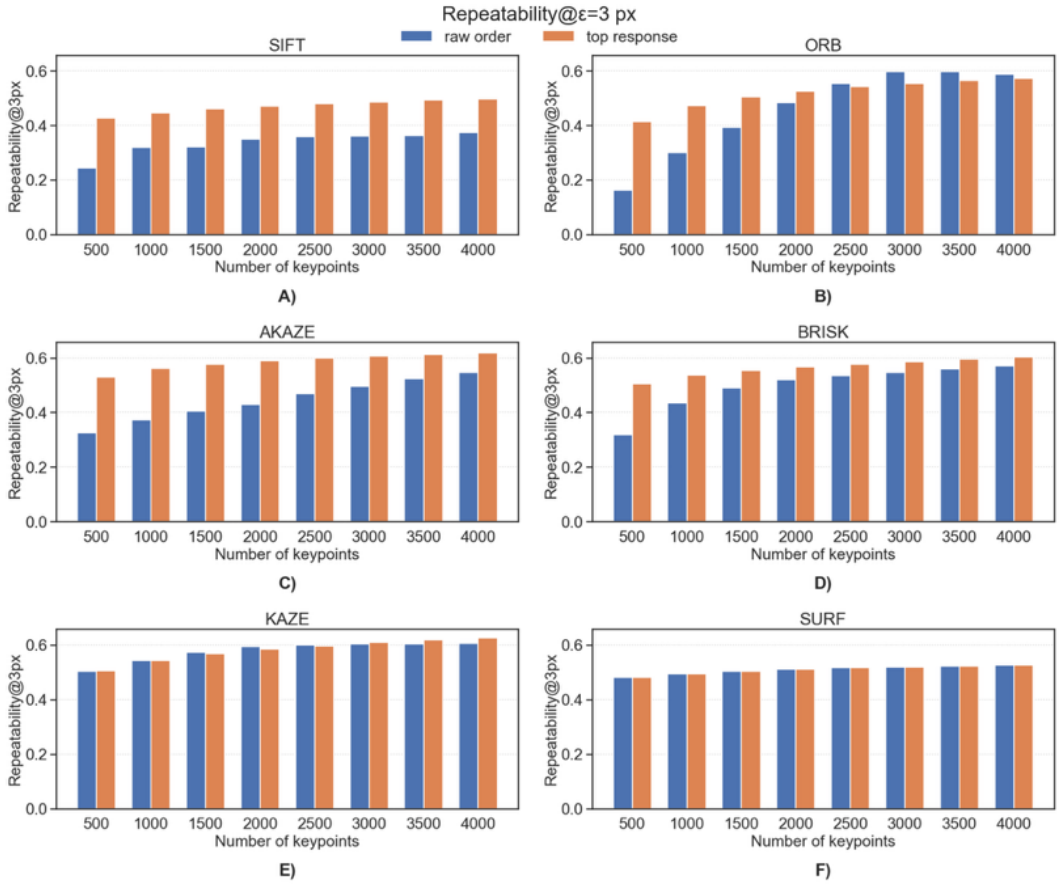


Fig. 8. Average Repeatability vs. the number of keypoints across detectors under two selection strategies: A) SIFT; B) ORB; C) AKAZE; D) BRISK; E) KAZE; F) SURF.

Overall, KAZE demonstrates the highest repeatability. Additionally, top_response primarily improves repeatability for SIFT, AKAZE, and BRISK at low keypoint counts, without altering the overall ranking of the methods.

Fig. 9 shows the dependencies of the Verification Ratio (VR). For most detectors, VR rises with the number of keypoints in the raw_order mode, while top_response consistently yields higher values even at small *N*, causing the curves to saturate early.

The highest VR values are observed for KAZE and SIFT in the top_response mode, at approximately 0.21-0.23. AKAZE also exhibits a comparable top_response of approximately 0.2, whereas in raw_order, the values are noticeably lower. For BRISK, the effect of selection by response is particularly pronounced at small *N* values: VR increases from about 0.09 to 0.17, after which the difference between the selection methods gradually decreases as *N* increases. ORB remains the least effective according to the VR metric; even in top_response, values are around 0.11–0.12, indicating a smaller proportion of points supporting a common homography under the conditions of various HPatches scenes. SURF stands out, with VR values remaining relatively high, though they show a moderate downward trend as the number of keypoints increases, from approximately 0.20 to 0.17. Overall, the best results under top_response are achieved by KAZE and AKAZE, whereas ORB exhibits the lowest proportion of geometrically verified points.

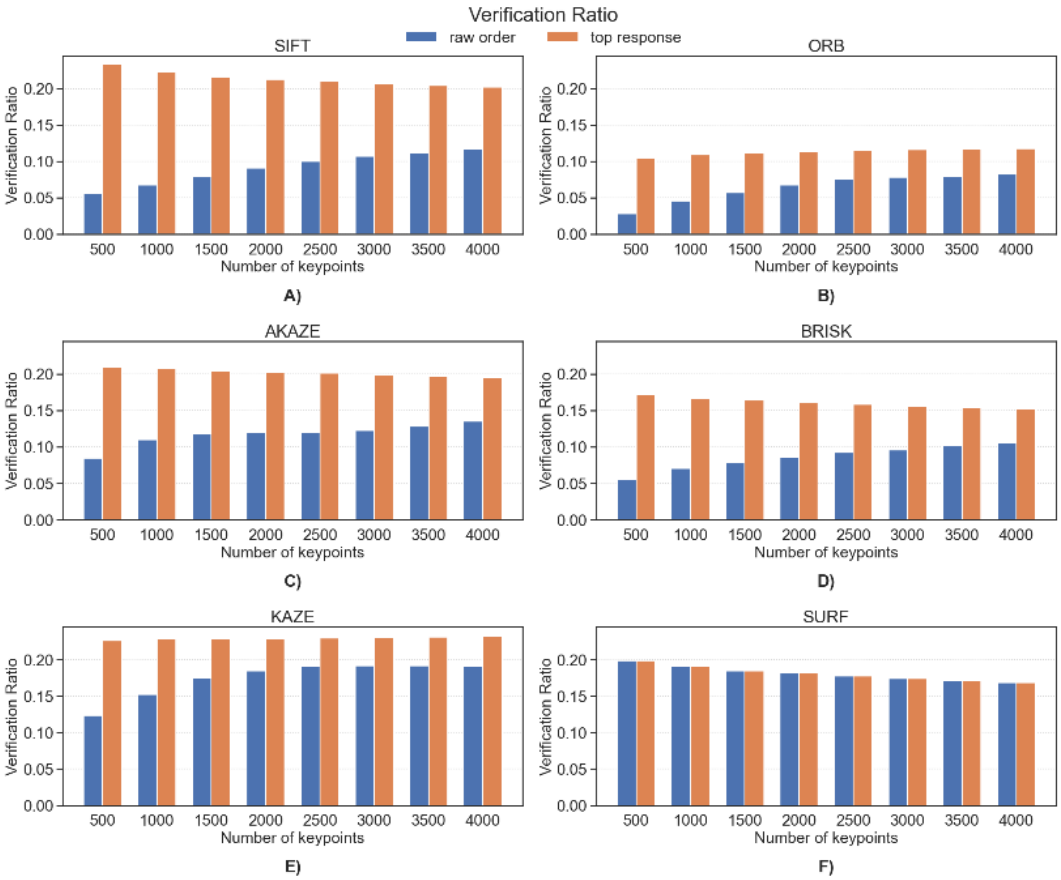


Fig. 9. Average Verification Ratio vs. the number of keypoints across detectors under two selection strategies: A) SIFT; B) ORB; C) AKAZE; D) BRISK; E) KAZE; F) SURF.

Fig. 10 illustrates the dependencies of the Quality Index, which includes the geometric and spatial components. The overall trend is moderate: the influence of the number of keypoints on the Q value is less significant than that of the detector choice and the point selection strategy.

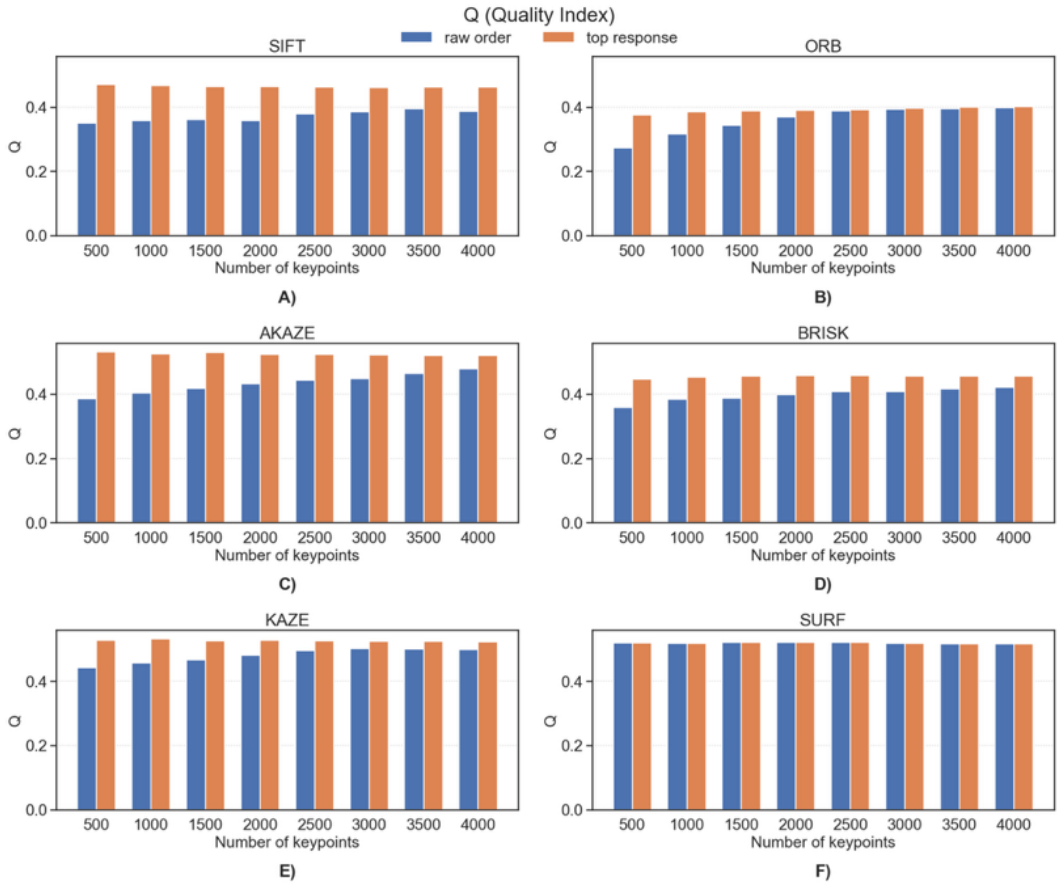


Fig. 10. Average Quality Index vs. the number of keypoints across detectors under two selection strategies: A) SIFT; B) ORB; C) AKAZE; D) BRISK; E) KAZE; F) SURF.

The highest Q values are achieved by the KAZE and AKAZE methods in the top_response mode, with averages of 0.526 and 0.524, respectively. The next best method is SURF, with an accuracy of approximately 0.51. For SIFT, the transition from raw_order to top_response yields a substantial increase in Q, from approximately 0.37 to 0.46, consistent with the simultaneous improvement in geometric and spatial characteristics in this mode. BRISK achieves an intermediate quality level of approximately 0.4 for raw_order and 0.45 for top_response. In contrast, ORB has the lowest Q values of approximately 0.36 and 0.39, respectively, reflecting limitations in both the proportion of geometrically verified correspondences and instructural-spatial metrics.

To verify the consistency of the Q, Principal Component Analysis (PCA) [28] was conducted on the MMA, Repeatability, VR, CUI_{filt}, SCS_{filt}, and RI_{filt} metrics. The first principal component, PC1, accounted for approximately 56% of the total variance and had positive loadings of similar magnitude across all measures, with the largest contributions from VR, CUI, and MMA. The correlation between Q and PC1 was approximatelyFinal_5_Article_text $r = 0.91$, suggesting that the proposed index effectively captures the main latent "axis" of quality derived by the principal component

method. This further confirms the correctness of the selected metrics and their weights in the *Q* formula.

The results obtained show that evaluating detectors solely on standard geometric metrics does not always capture differences in the spatial organization of keypoints, which significantly affect the suitability of the correspondence set. Spatial metrics reveal additional patterns: an increase in geometric performance may be accompanied by changes in coverage uniformity, local redundancy, and the structural consistency of keypoints with the scene; therefore, for a comprehensive comparison, it is advisable to consider both groups of characteristics together. This focus on spatial structure aligns with modern approaches that examine keypoint stability and quality through clustering characteristics and spatial patterns; relevant studies highlight the importance of assessing detector properties beyond matching accuracy [19-21, 29].

The analysis of point selection strategies is also fundamental [30]. An exception is the SURF detector, where both strategies yielded nearly identical results. This occurs due to the specifics of the OpenCV implementation, which returns keypoints already sorted by the response magnitude linked to the Hessian measure 17].

The proposed metric *Q* enables comparison of detectors while accounting for the factors mentioned earlier. PCA was used as an initial validation tool and verified that the *Q* index aligns with the dominant direction of variability in the normalized metrics. A more detailed analysis of the component structure and loading stability is planned for future work.

CONCLUSION

This study offers a comprehensive evaluation of the SIFT, SURF, ORB, BRISK, KAZE, and AKAZE keypoint detectors on the HPatches dataset, using USAC to validate matches geometrically. It shows that assessing methods solely based on traditional geometric metrics - such as MMA, Repeatability, and Verification Ratio - does not fully capture the differences among detectors because it overlooks the spatial arrangement of keypoints. The proposed spatial-structural metrics, CUI, RI, and SCS, provide a quantitative description of frame coverage uniformity, local redundancy, and the consistency of keypoint distribution with scene structure, thereby enhancing standard accuracy analyses.

The results showed that the selection strategy had a greater impact on detector behavior than the number of keypoints. Selecting keypoints by response strength consistently increased MMA, Repeatability, and Verification Ratio across all methods, with the strongest effects observed for SIFT, ORB, and BRISK. At the same time, the analysis of spatial characteristics demonstrated that not all detectors consistently achieve uniform scene coverage and structural consistency. The most balanced CUI, RI, and SCS values were recorded for KAZE and AKAZE; high values were also obtained for SURF, whereas ORB and BRISK were found to be more specialized toward corner-like structures and tended to form redundant, clustered point sets.

The proposed Quality Index, which combines spatial and geometric metrics, enables the generalization of evaluation and comparison across detection methods. The KAZE and AKAZE detectors achieved the highest average *Q* values, with SURF ranked second. The high scores of these methods simultaneously ensure acceptable geometric accuracy, sufficient repeatability, and a spatial distribution of keypoints that is close to uniform. The performance of SIFT and BRISK was competitive in terms of *Q* values when keypoints were selected by response strength. The ORB method remained the least balanced according to the generalized *Q* indicator, despite significant improvements in matching accuracy after sorting keypoints by response.

Principal component analysis indicated that the first component effectively summarized the variability of the metrics and was strongly correlated with the Q index,

thereby providing additional validation of the chosen approach. This result suggests that the proposed indicator accurately aggregates inlier-set properties and can serve as a general criterion for detector comparison in computer-vision tasks where both homography accuracy and scene coverage are essential. In future work, the conclusions should be verified on additional datasets and geometric models – notably the fundamental matrix – and the analysis should be extended to a broader range of imaging conditions, including illumination changes.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author(s) received no financial support for the research, writing, and/or publication of this article.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [A.F., Yu.F.]; methodology, [A.F., Yu.F.]; validation, [A.F., Yu.F.]; writing – original draft preparation, [A.F.]; writing – review and editing, [A.F., Yu.F.]; supervision, [Yu.F.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Zhou, L., Wu, G., Zuo, Y., Chen, X., & Hu, H. (2024). A comprehensive review of vision-based 3D reconstruction methods. *Sensors, 24*(7), 2314. https://doi.org/10.3390/s24072314

[2] Ye, Z., Bao, C., Zhou, X., Liu, H., Bao, H., & Zhang, G. (2023). EC-SfM: Efficient covisibility-based structure-from-motion for both sequential and unordered images. *IEEE Transactions on Circuits and Systems for Video Technology*. https://doi.org/10.1109/TCSVT.2023.3285479

[3] Huang, Q., Guo, X., Wang, Y., Sun, H., & Yang, L. (2024). A survey of feature matching methods. *IET Image Processing, 18*(6), 1385-1410. https://doi.org/10.1049/ipr2.13032

[4] Herrera-Granda, E. P., Berrones-González, A., & Aguilar, W. (2024). Monocular visual SLAM, visual odometry, and structure from motion: A review. *Heliyon, 10*(9), e37356. https://doi.org/10.1016/j.heliyon.2024.e37356

[5] Al-Tawil, Y., Zagraoui, A., & Zeghbib, A.-R. (2024). A review of visual SLAM for robotics: evolution, properties, and future applications. *Frontiers in Robotics and AI, 11*, 1347985. https://doi.org/10.3389/frobt.2024.1347985

[6] Ehsan, S., Kanwal, N., Clark, A. F., & McDonald-Maier, K. D. (2011). Measuring the coverage of interest point detectors. In M. Kamel & A. Campilho (Eds.), *Image Analysis and Recognition (ICIAR 2011)* (Lecture Notes in Computer Science, Vol. 6753, pp. 253-261). Springer. https://doi.org/10.1007/978-3-642-21593-3_26

[7] Mousavi, V., Varshosaz, M., & Remondino, F. (2021). Using information content to select keypoints for UAV image matching. *Remote Sensing, 13*(7), 1302. https://www.mdpi.com/2072-4292/13/7/1302

[8] Balntas, V., Lenc, K., Vedaldi, A., & Mikolajczyk, K. (2017). HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2017.410

[9] Balntas, V., Lenc, K., Vedaldi, A., Tuytelaars, T., Matas, J., & Mikolajczyk, K. (2020). H-Patches: A benchmark and evaluation of handcrafted and learned local

descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(11), 2825-2841. https://doi.org/10.1109/TPAMI.2019.2915233
[10] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
[11] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding, 110*(3), 346-359. https://doi.org/10.1016/j.cviu.2007.09.014
[12] Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012). KAZE features. In *ECCV 2012* (LNCS 7577, pp. 214-227). https://doi.org/10.1007/978-3-642-33783-3_16
[13] Alcantarilla, P. F., Nuevo, J., & Bartoli, A. (2013). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *BMVC 2013* (pp. 1-11). https://doi.org/10.5244/C.27.13
[14] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *ICCV 2011* (pp. 2564-2571). https://doi.org/10.1109/ICCV.2011.6126544
[15] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. In *ICCV 2011* (pp. 2548-2555). https://doi.org/10.1109/ICCV.2011.6126542
[16] M. Ivashechkin, D. Baráth, J. Matas, "USACv20: Robust Essential, Fundamental and Homography Matrix Estimation," 2021. https://doi.org/10.48550/arXiv.2104.05044
[17] Howse, Joseph, and Joe Minichino. “Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning.”, *Packt Publishing Ltd*, 2020.
[18] Fesiuk, A., & Furgala, Y. (2025). Keypoint matches filtering in computer vision: Comparative analysis of RANSAC and USAC variants. *International Journal of Computing, 24*(2), 343-350. https://doi.org/10.47839/ijc.24.2.4018
[19] Hossein-Nejad, Z., & Nasri, M. (2017). RKEM: Redundant keypoint elimination method in image registration. *IET Image Processing, 11*, 273-284. https://doi.org/10.1049/iet-ipr.2016.0440
[20] Hossein-Nejad, Z., Agahi, H., & Mahmoodzadeh, A. (2021). Image matching based on the adaptive redundant keypoint elimination method in the SIFT algorithm. *Pattern Analysis and Applications, 24*(2), 669-683. https://doi.org/10.1007/s10044-020-00938-w
[21] Hossein-Nejad, Z., & Nasri, M. (2022). Clustered redundant keypoint elimination method for image mosaicing using a new Gaussian-weighted blending algorithm. *The Visual Computer, 38*(6), 1991-2007. https://doi.org/10.1007/s00371-021-02261-9
[22] Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference* (pp. 147-151). https://doi.org/10.5244/C.2.23
[23] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8*(6), 679-698. https://doi.org/10.1109/TPAMI.1986.4767851
[24] Soille, P. (2004). *Morphological Image Analysis: Principles and Applications* (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-05088-0
[25] Rey-Otero, I., Delbracio, M., & Morel, J.-M. (2015). Comparing feature detectors: A bias in the repeatability criteria. In *2015 IEEE International Conference on Image Processing (ICIP)* (pp. 3024-3028). https://doi.org/10.1109/ICIP.2015.7351358
[26] Rey-Otero, I., & Delbracio, M. (2015). Is repeatability an unbiased criterion for ranking feature detectors? *SIAM Journal on Imaging Sciences, 8*(4), 2558-2580. https://doi.org/10.1137/15M1007732

[27] Avriel, M., & Wilde, D. J. (1968). Golden block search for the maximum of unimodal functions. *Management Science, 14*(5), 307-319 https://doi.org/10.1287/mnsc.14.5.307
[28] Boudt, K., d'Errico, M., Luu, H. A., & Pietrelli, R. (2022). Interpretability of Composite Indicators Based on Principal Components. *Journal of Probability and Statistics*, 2022**,** 4155384, 1-12. https://doi.org/10.1155/2022/4155384
[29] Mukherjee, S., Lagache, T., & Olivo-Marin, J.-C. (2021). Evaluating the stability of spatial keypoints via cluster core correspondence index. *IEEE Transactions on Image Processing, 30*, 386-401. https://doi.org/10.1109/TIP.2020.3036759
[30] A. Fesiuk and Y. Furgala, (2025). Comparative Study of Feature Detectors and Keypoint Filters in Image Matching. *Electronics and Information Technologies*, (31), (pp. 71-88). http://dx.doi.org/10.30970/eli.31.7

КОМПЛЕКСНЕ ПРОСТОРОВО-ГЕОМЕТРИЧНЕ ОЦІНЮВАННЯ ДЕТЕКТОРІВ КЛЮЧОВИХ ТОЧОК

Андрій Фесюк, **Юрій Фургала***
Факультет електроніки та комп'ютерних технологій
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 Львів, Україна

АНОТАЦІЯ

Вступ. Локальні ознаки є важливими компонентами сучасних систем комп'ютерного зору, таких як SLAM та 3D-реконструкція. Традиційні підходи до оцінювання методів детекції зосереджуються переважно на геометричній точності та повторюваності, часто не враховуючи просторову структуру розподілу точок. Це ускладнює вибір алгоритму для задач, де важливими є рівномірність покриття кадру та відсутність надлишкової локальної кластеризації. Метою роботи є комплексне порівняння детекторів із використанням розширеного набору метрик, що враховують як геометричну коректність, так і просторові властивості ознак.

Матеріали та методи. Дослідження проведено на наборі даних HPatches для шести детекторів: SIFT, SURF, ORB, BRISK, KAZE, AKAZE. Фільтрація особливих точок виконана методом USAC. Якість зіставлення оцінено за геометричними метриками MMA, Repeatability, Verification Ratio. Для просторового аналізу використано метрики CUI, RI та SCS. Для узагальненого порівняння методів детекції ключових точок запропоновано індекс якості Q, який об'єднує геометричні та просторові показники.

Результати. Результати показали, що стратегія відбору точок за силою відгуку суттєво підвищує точність зіставлення для SIFT, ORB та BRISK, проте призводить до локальної надлишковості ключових точок. Результати методів KAZE та AKAZE продемонстрували найкращий баланс, забезпечуючи високу точність при рівномірному покритті сцени. Метод ORB виявився схильним до формування щільних скупчень у контрастних зонах, що знижує його структурну ефективність, тоді як SURF показав стабільно високі результати незалежно від стратегії відбору ключових точок.

Висновки. Запропонований підхід до оцінювання забезпечує узгоджений аналіз геометричних і просторових властивостей детекторів ключових точок та показує, що за фіксованої кількості ключових точок на підсумкову якість методу істотно впливають не лише показники геометричної коректності відповідностей, а й характеристики просторового розподілу точок. Встановлено, що спосіб вибору особливих точок,

зокрема відбір точок за силою відгуку, систематично змінює як геометричні, так і просторові властивості. Індекс якості Q узагальнює ці аспекти в єдиному показнику та може застосовуватися для порівняння методів детекції у сценаріях, де потрібні одночасно надійні відповідності та збалансоване покриття сцени.

Ключові слова: виявлення ознак, просторовий розподіл, геометричні метрики, співпадіння.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
10 December, 2025	17 December, 2025	17 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 87–100
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.8,004.9

EXPLAINABLE AND INTERPRETABLE MACHINE LEARNING MODELS FOR ANALYSIS OF OPEN BANKING DATA

Markiyan Fostyak , ***Lidiia Demkiv****
System Design Department
Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine

Fostyak M., Demkiv L. (2025). Explainable and Interpretable Machine Learning Models for Analysis of Open Banking Data. *Electronics and Information Technologies*, 32, 87–100. https://doi.org/10.30970/eli.32.6

ABSTRACT

Background. The development of artificial intelligence and machine learning models has significantly influenced financial analytics and credit decision-making. These models provide high predictive accuracy but often operate as “black boxes,” which complicates the interpretation of their internal mechanisms. In the context of open banking, where decisions directly affect users’ access to financial resources, such opacity is a substantial drawback. This creates a need for explainable and interpretable approaches that make it possible to establish causal relationships between input features and output predictions.

Materials and Methods. The research methods are based on a multi-level approach to ML model interpretation. Feature Importance is applied for a statistical assessment of feature contributions; LIME is used to provide local interpretability; and SHAP (SHapley Additive exPlanations) is employed to capture nonlinear dependencies. Structural interpretability is ensured by DNFS (Deep Neuro-Fuzzy System) through the formation of fuzzy rules, while BRB-ER (Belief Rule Base with Evidential Reasoning) adds logically consistent explanations of decisions based on a rule base.

Results and Discussion. It is shown that, after hyperparameter optimization of credit risk models trained on open banking data, the accuracy of the DNFS model becomes 4 percentage points higher than that of the Random Forest model. A global analysis of feature importance scores obtained using Feature Importance, SHAP, and DNFS demonstrates a high correlation between them (above 88%), indicating model stability. At the local level, instances that reduce model accuracy are identified. Visualizations using SHAP graphs reveal regions of linear and nonlinear feature interactions and their influence on decision-making.

Conclusion. In contrast to the traditional use of individual XAI methods to explain machine learning model outputs, this work combines global and local feature importance metrics (Feature Importance, SHAP, LIME), fuzzy rule–based metrics from DNFS, and aggregated coefficients from BRB-ER. The proposed approach makes it possible to localize the causes of accuracy degradation, identify nonlinear feature dependencies, and assess the consistency of explanations through correlation analysis across methods.

Keywords: explainable artificial intelligence, machine learning, BRB, DNFS, fuzzy logic.

INTRODUCTION

Over the past few years, machine learning (ML) models have been widely used to solve a variety of tasks. However, the limited transparency of their results creates obstacles to practical deployment, especially in domains where the explainability of outcomes is of

© 2025 Markiyan Fostyak & Lidiia Demkiv. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

critical importance. This primarily concerns medicine, financial analytics, and decision support systems, where trust in algorithmic predictions directly depends on the ability to interpret their internal logic. For this reason, there is growing interest in such areas as Explainable Artificial Intelligence (XAI) and the development of interpretable models. Reviews [1–3] present more than ten methods used to construct explainable models in the financial domain. In finance, XAI has the highest priority due to the complex and often opaque structure of ML models used for credit rating assessment, bankruptcy prediction, fraud detection, and credit portfolio optimization.

To explain such models, the following methods are commonly employed: LIME (Local Interpretable Model-agnostic Explanations), which builds local linear approximations to explain the behavior of complex models in the vicinity of an individual prediction; SHAP (SHapley Additive exPlanations), a method based on Shapley value theory that decomposes the model output into contributions of each feature according to its marginal importance; TreeSHAP or TreeExplainer, a modification of SHAP optimized for tree-based models (Random Forest, XGBoost), which provides exact analytical values without the need for simulations; PDP (Partial Dependence Plot), a graphical tool that visualizes the average effect of one or several features on the model prediction; Anchors (High-Precision Model-Agnostic Explanations), a technique that constructs interpretable “anchor” rules providing high-precision local explanations; Counterfactual Explanations (CF), which identify minimal changes in the input data that would lead to an alternative model output; Permutation Feature Importance (PFI), a simple global metric that measures the drop in model accuracy after random permutation of a specific feature; and Surrogate Models (SM), which approximate the behavior of a complex model with an interpretable structure, such as a decision tree or linear regression, among others.

The theoretical foundations and prospects of Explainable Artificial Intelligence (XAI) are discussed in [4,5]. The practical application of XAI in various types of financial models is examined in [6–9]. All these studies confirm that integrating XAI approaches into financial analytics increases trust in automated systems and helps maintain a balance between model accuracy and interpretability. In [6], the SHAP method is used to explain the results of a LightGBM model in a credit scoring task, which made it possible to identify key factors influencing loan approval decisions. In [7], XGBoost, LightGBM, and Random Forest algorithms are applied for predictive analysis of loan default risk in combination with XAI methods SHAP and LIME. Study [8] presents a method for credit card default prediction based on a combination of deep learning and explainable artificial intelligence (XAI) techniques.

Another direction in the development of explainable artificial intelligence is hybrid interpretable approaches. One such approach is the Belief Rule Base with Evidential Reasoning (BRB-ER). BRB-ER ensures transparency and interpretability of decision-making, since each prediction is formed based on an intelligible system of logical rules. In [9], the hierarchical BRB structure incorporates both factual and heuristic rules and can explain the chain of events leading to a decision on a loan application. In addition, performance indicators for evaluating the rule base are proposed, including quantitative, qualitative, and visual metrics in the form of a rule interaction graph [9,10]. Another line of development for interpretable models is Deep Neuro-Fuzzy Systems (DNFS). In DNFS, the hidden layers implement the fuzzification process, creating fuzzy sets for input features, after which the system automatically generates and tunes “if–then” rules that reflect causal relationships in the data [11,12]. Unlike BRB, DNFS does not require prior specification of expert rules, but instead learns them during training, which provides data-driven adaptive interpretability. The importance of each feature can be computed based on the activation strength of the corresponding rules or on the weights of neural connections.

This study aims to compare the effectiveness of interpreting ML models using BRB-ER and DNFS approaches with XAI methods, in particular SHAP and LIME, in terms of their ability to determine feature importance, and to define a consistent approach to

constructing explainable and interpretable models for financial analysis in open banking systems.

MATERIALS AND METHODS

Explaining and interpreting the results of machine learning models, as well as analyzing the characteristics of the data on which these models are trained, requires a combination of classical XAI methods, hybrid neuro-fuzzy systems, and rule-based logical approaches. To this end, this study implements an integration of global and local XAI techniques with hybrid approaches and fuzzy logic, which makes it possible to analyze both the overall patterns captured by the model and individual decisions for specific instances.

The SHAP (SHapley Additive exPlanations) method quantifies how important each input variable is for the model. SHAP values are derived from Shapley values in cooperative game theory, where each feature is treated as a “player” contributing to the collective outcome, i.e., the model prediction. The key idea behind SHAP is that the Shapley value is expressed as an additive feature attribution method via a linear explanation model.

In the Python implementation of SHAP, a specialized algorithm called TreeSHAP is provided, which is optimized for tree-based ML models, in particular Decision Trees, Random Forests, and Gradient Boosted Trees (XGBoost, LightGBM, CatBoost). TreeSHAP is a computationally efficient variant of SHAP that does not compute Shapley values via exhaustive enumeration of all feature subsets, but instead relies on an analytical analysis of the tree structure [13]. The algorithm exploits the fact that a tree prediction is a deterministic function of the split conditions at its internal nodes. By analyzing all possible decision paths, it is therefore possible to determine exactly which features actually influence the model output. When SHAP values are used to explain real-world outcomes, it must be recognized that SHAP only shows what the model does in the context of the data on which it was trained. The method does not necessarily uncover the true causal relationships between variables and outcomes in the real world.

The LIME (Local Interpretable Model-agnostic Explanations) method implements the concept of local interpretability by constructing a simplified local regression model (typically linear) that approximates the predictions of the base model in the immediate neighborhood of a selected instance for a subset of points close to the observation of interest. To construct such an explanation, LIME generates a synthetic local dataset by randomly sampling points around the instance, and then fits a linear regression model whose coefficients are used as local feature attributions. The effectiveness of LIME depends strongly on the choice of its parameters: the kernel width defining the locality, the number of generated neighboring points, and the regularization parameter controlling the number of features in the local model [14]. At the same time, the fidelity of the explanation depends on the adequacy of the local linear approximation: if the global model exhibits strong nonlinearity or complex feature interactions, the fitted linear model may capture only a very small, highly local fragment of its behavior.

A separate line of XAI research concerns the construction of hybrid solutions in which interpretable or explainable models are combined with optimization algorithms. In hybrid systems of the type ML + BRB + Optimizer, the ML component is responsible for predictive accuracy, the BRB component ensures transparency via a rule base, and the optimization algorithm adjusts the structure and parameters of the rules to align them with the outputs of the “black-box” model [15]. Such an approach is used, for example, in credit scoring, where predictive accuracy must be combined with the ability to explain the result to the end user and to regulators. In [16], we described in detail a hybrid decision-support system that combines a Belief Rule Base (BRB) model, a machine learning (ML) model, and Particle Swarm Optimization (PSO). At the first stage, reference values of the input features are defined to represent the linguistic states of the parameters. At the second stage, based on these reference values, a BRB rule base is constructed, where each rule encodes the

relationship between a combination of input attribute states and the corresponding class. Each rule is characterized by a rule weight and attribute weights. The Evidential Reasoning (ER) mechanism is used to aggregate the activated rules into a final output. At the third stage, the machine learning model (ML) is trained. At the final stage, Particle Swarm Optimization (PSO) is applied to automatically tune the parameters. Analyzing the parameters learned makes it possible to identify which rules or attributes have the greatest influence on the result. Consequently, the BRB–ML–PSO system produces not only a numerical output, but also a logical interpretation of this output via the rule base.

Deep Neuro-Fuzzy System (DNFS) is a hybrid approach that combines fuzzy rules (fuzzy logic) with deep neural networks (deep learning). Unlike BRB, DNFS does not explicitly store all possible rules. Instead, a neural network automatically learns the relevance of rules, prunes insignificant combinations (through the weights), and generalizes, so that the effective number of rules does not grow exponentially.

Structurally, DNFS consists of several interconnected layers [17]. At the first level, each feature passes through a separate fuzzification layer, where it is transformed into fuzzy terms representing the linguistic states of the variable. This is implemented via sigmoidal activation functions. As a result, fuzzy features are formed that represent fuzzy intervals of the values of each input variable. The generated fuzzy features are concatenated into a vector in a fuzzy feature space, which is fed into the rule layer, a layer of neurons. Each neuron corresponds to a single fuzzy IF–THEN rule, and its weights determine the importance of individual terms in the rule combination. During training, the parameters of the fuzzy terms and the rule weights are optimized using a gradient-based method (the Adam optimizer). After computing rule activation degrees, they are normalized using the Softmax function, which transforms them into a distribution of influence weights over the rules. The final output layer aggregates these normalized rule contributions and, in the case of classification, again applies Softmax to produce a probabilistic class prediction.

As a result, we obtain an interpretable output in the form of rules that are generated automatically based on the network's learned parameters, as well as rule activation degrees and normalized rule weights. Feature importance in DNFS is computed from the aggregate influence of the weights connecting fuzzy terms to the rule layer. Since each feature is represented by several terms, its integral importance is defined as the sum of the absolute weights linking its terms to all rules. This approach makes it possible to quantify the contribution of each feature to the system's decisions and to identify which rules determined the outcome and why. In this way, the learned parameters can be used for explainability (XAI), analogous to SHAP or BRB. Thus, the DNFS architecture combines the high predictive accuracy of a neural model with the interpretability of fuzzy logic, which makes it effective for financial and medical applications where not only predictive accuracy but also an explanation of the underlying reasons is crucial.

With the development of financial technologies (FinTech) and the digitalization of banking services, the role of ML models in decision-making related to lending, risk assessment, and forecasting clients' financial behavior has increased significantly. One of the modern approaches to building credit rating models is the use of open banking API data, which contains detailed information on client transactions. At the data preparation stage, the raw open banking API data are preprocessed and transformed into features such as income stability (SI), mandatory spending (MS), non-mandatory (discretionary) spending (DS), risky spending (RS), and account balance [18,19]. To reduce the total number of BRB rules, further categorization of financial features is performed, which we described in detail in [16]. As a result, four features are obtained and expressed in relative units: stability of income (SI), stability of discretionary spending (SDS = 1 − SD), stability of risky spending (SRS = 1 − RS), and the difference between income and expenditure (Diff)

Achieving high classification accuracy on open banking data is a challenging task due to the nature of transactional data itself [20], which is characterized by high variability and

low correlation between individual features and the target variable (Fail). Here, Fail = 1 corresponds to the highest credit risk level. Analysis of the correlation matrix shows that the maximum correlation of 0.52 is observed between Diff and Fail, while all other correlation values between variables are below 0.40.

All experiments were implemented in Python 3.8.6. The following libraries were used: NumPy [1.24.3], pandas [1.5.3], matplotlib [3.7.5], scikit-learn [1.3.2], SHAP [0.44.1], LIME [0.2.0], and TensorFlow/Keras [2.13.0]. In addition to using these established packages for model training and explainability, custom Python scripts were implemented for the core study workflow, including: data loading from CSV and preprocessing (feature extraction/ordering for SI, SDS, SRS, DiFF and label handling), training and evaluation of a Random Forest classifier (train/test split, confusion matrix, accuracy, classification report, ROC-AUC, and partial dependence plots), generation of explanation artifacts (SHAP global summaries and dependence plots, SHAP waterfall plots for individual instances, and LIME local surrogate explanations), implementation of a Belief Rule Base inference engine comprising triangular fuzzification (Low/Medium/High on [0,1]), a complete rule base with belief degrees and optional rule weights, rule activation and belief aggregation using weighted averaging and an Evidential Reasoning style combination, and development of a DNFS (Deep Neuro-Fuzzy System) model in TensorFlow/Keras, including fuzzification layers, a rule layer, softmax-based normalization, and rule extraction from learned weights for interpretability.

RESULTS AND DISCUSSION

Table 1 presents the classification accuracy of the Random Forest (RF) and Deep Neuro-Fuzzy System (DNFS) models before and after hyperparameter optimization. The analysis is performed on open banking data containing four key financial parameters of clients (SD, SDS, SRS, Diff) and a binary target variable (Fail) that characterizes the level of credit risk. Among the ensemble machine learning models considered — Gradient Boosting (XGBClassifier), AdaBoostClassifier, and CatBoostClassifier — the Random Forest model demonstrated the highest classification accuracy.

Particle Swarm Optimization (PSO) was used to determine the optimal hyperparameter values for both RF and DNFS. For RF, the following hyperparameters were optimized: the number of trees (n_estimators), maximum tree depth (max_depth), minimum number of samples required to split an internal node (min_samples_split), and minimum number of samples at a leaf node (min_samples_leaf). For DNFS, the optimized parameters included the number of terms per feature (terms_per_feature), the number of rules (number_of_rules), the learning rate, and the dropout rate.

Table 1. **Accuracy of Random Forest (RF) and Deep Neuro-Fuzzy System (DNFS) models before and after the optimization procedure**

	RF	RF (opt)	DNFS	DNFS (opt)
Accuracy	0.84	0.88	0.83	0.91

As can be seen from **Table 1**, both models exhibit an increase in accuracy after optimization. The DNFS model shows a more pronounced improvement in accuracy, which can be attributed to the structural alignment of its rules with the data.

Global explanation

Table 2 presents the feature importance coefficients for the variables SD, SDS, SRS, and Diff, computed using three different approaches (Feature Importance, SHAP, DNFS). The Feature Importance coefficients capture the global statistical contribution of each feature to classification accuracy over the entire dataset. They are obtained from the

Random Forest model using the feature_importances_ attribute, which calculates the mean decrease in impurity (Gini importance) across the decision trees.

Global SHAP coefficients are derived by aggregating local SHAP values for each data instance using shap.Explainer(model), which makes it possible to estimate the average effect of each feature on the model prediction. Global feature importance in DNFS is determined based on the weights of fuzzy terms in the learned rules. Example rules for different classes have the following form:

- Rule 4: IF Diff_Low (w = −0.69) AND SI_High (w = −0.54) AND SD_High (w = −0.45) AND SI_Low (w = 0.35) AND SR_High (w = 0.31) AND SR_Low (w = 0.24) THEN → Class 1;
- Rule 5: IF Diff_High (w = 0.56) AND Diff_Low (w = 0.47) AND SI_Low (w = 0.42) AND SI_High (w = 0.40) AND SD_High (w = 0.39) AND SD_Low (w = −0.39) THEN → Class 0.

For each rule, the absolute values of the weights of the terms associated with a given feature were summed, and these sums were then aggregated across the entire rule set. Subsequent normalization of the resulting values made it possible to obtain an integral rule-based estimate of the importance of each attribute. The resulting aggregate coefficients reflect the degree of influence of each feature on the final model decision.

The DNFS architecture consists of four sequential layers: the input layer, the fuzzy layer (12 neurons, sigmoid), the rule layer (45 neurons, sigmoid), and the decision layer (2 neurons, softmax). The results show that the obtained DNFS-based importance coefficients are highly correlated with the global SHAP values, which confirms the methodological consistency of the different levels of explanation.

A three-component Pearson correlation matrix was computed for the feature importance coefficients. The high correlation between the results of all three approaches (greater than 0.88) indicates that the overall pattern of how influence is distributed across features remains consistent.

Table 2. **Global feature importance coefficients**

	Feature importance	SHAP	DNFS
Diff	0.56	0.36	0.37
SRS	0.19	0.08	0.23
SDS	0.17	0.05	0.21
SI	0.06	0.02	0.19

For a more detailed interpretation of the global impact of the features, a beeswarm plot (**Fig. 1**) was constructed to visualize the distribution of local SHAP values for each feature. The Y-axis lists the features in descending order of their SHAP-based importance, while the X-axis shows the SHAP values, which reflect the direction and magnitude of their influence on the probability of the class Fail = 1. Each point corresponds to an individual data instance, and its color encodes the actual value of the feature. The plot clearly shows that low values of all variables (except for SDS) increase credit risk in a more linear manner and much more rapidly than high values of these variables. For high values of all features, the interactions are more strongly nonlinear than for low values.

The feature SDS exhibits a strongly nonlinear effect across its entire range of values. The atypical influence of SDS (discretionary spending) on credit risk at low spending levels is associated with the borrower's limited ability to redirect these expenditures towards loan repayment.

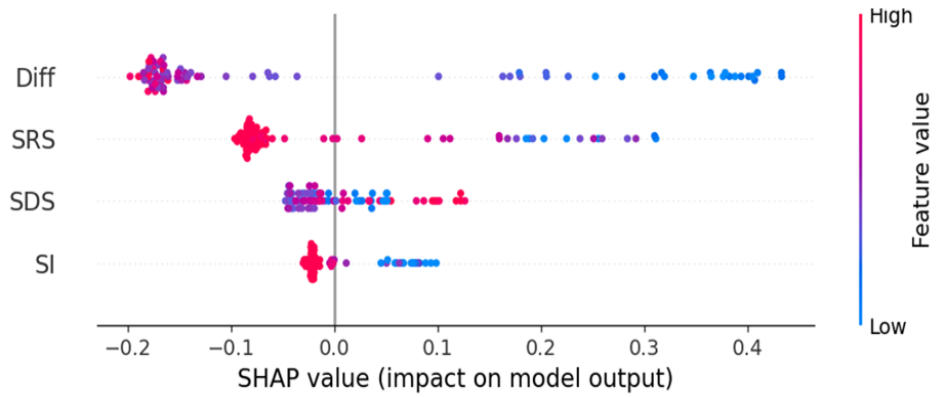


Fig. 1. Distribution of local SHAP values for each feature.

To further assess the global behavior of the model, Partial Dependence Plots (PDPs) (**Fig. 2**) were constructed for the four input features. PDPs make it possible to visually examine how changes in the value of a single feature affect the model's prediction while holding the remaining variables at their average levels. The X-axis represents the values of the corresponding feature, and the Y-axis shows the mean predicted value of the target variable, averaged over all observations.

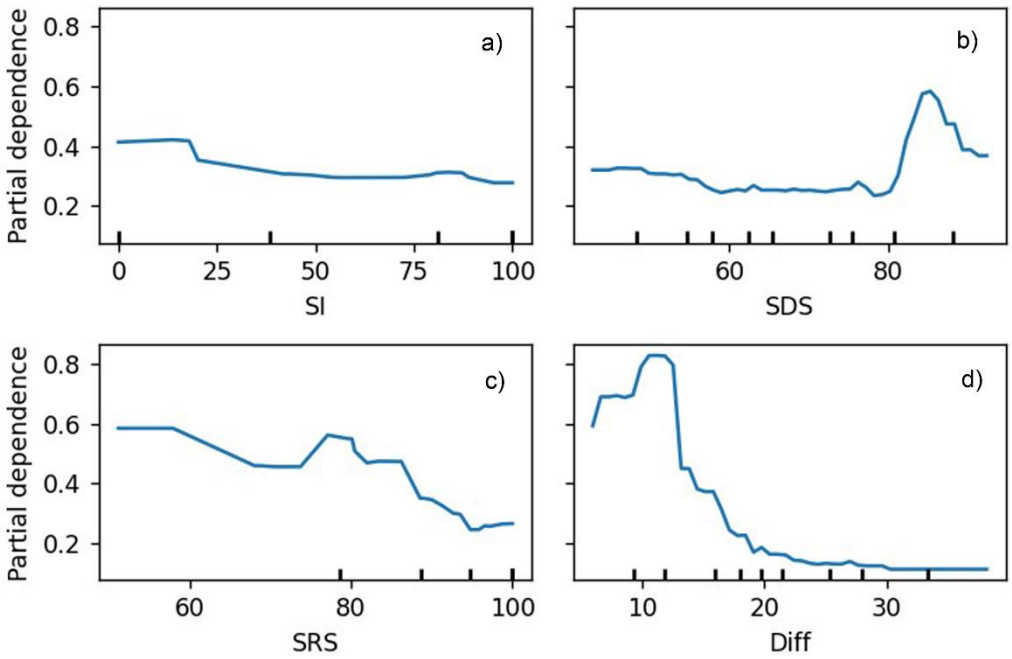


Fig. 2. Partial Dependence Plot: a – Partial Dependence–Si, b – Partial Dependence–SDS, c – Partial Dependence–SRS, d – Partial Dependence–Diff.

As income stability (SI) and stability of risky spending risky spending (SRS) increases, an almost smooth reduction in credit risk is observed. For SDS, two distinct regions of small and large values (i.e., very low and very high discretionary spending) can be seen, both of which are associated with an increase in credit risk. For the difference between income and expenditure (**Fig. 2d**), two distinct dependence regions are observed. For Diff > 12, credit

risk decreases in an almost exponential manner up to a point beyond which further increases in the income–expenditure gap no longer improve the risk level. This behavior reflects a zone of financial stability, beyond which additional resources do not increase the probability of loan approval.

For Diff < 12, a threshold-type relationship between Diff and credit risk is observed, namely, a decrease in risk as the difference decreases. Such atypical behavior may be driven by various underlying factors, which require more detailed investigation at the local level.

Local explanation

To conduct a detailed analysis of the local, instance-specific effects of features on the model's predictions, the Individual Conditional Expectation (ICE) method was employed. In contrast to global dependencies (**Fig. 2**), the ICE plot in **Fig. 3** depicts the trajectory of the model's predicted value for a single observation when only one selected feature is varied, while all other characteristics are held fixed. One of the characteristics of the average ICE curve is its mean slope. The largest mean slope, 0.023, corresponds to the feature Diff (**Fig. 3d**).; identical mean slopes of 0.019 are observed for the SI (**Fig. 3a**) and SRS (**Fig. 3c**) features, and the smallest mean slope, 0.010, is obtained for the SDS feature (**Fig. 3b**).

The application of ICE in the context of credit risk assessment made it possible to detect local nonlinearities that are not apparent in the PDPs (**Fig. 2**) or in the average curve (**Fig. 3**). The presence of an S-shaped structure in the ICE plot for Diff indicates that even small increases or decreases in Diff substantially change the level of credit risk. This behavior suggests that the model responds to this feature in a non-linear, threshold-like manner.

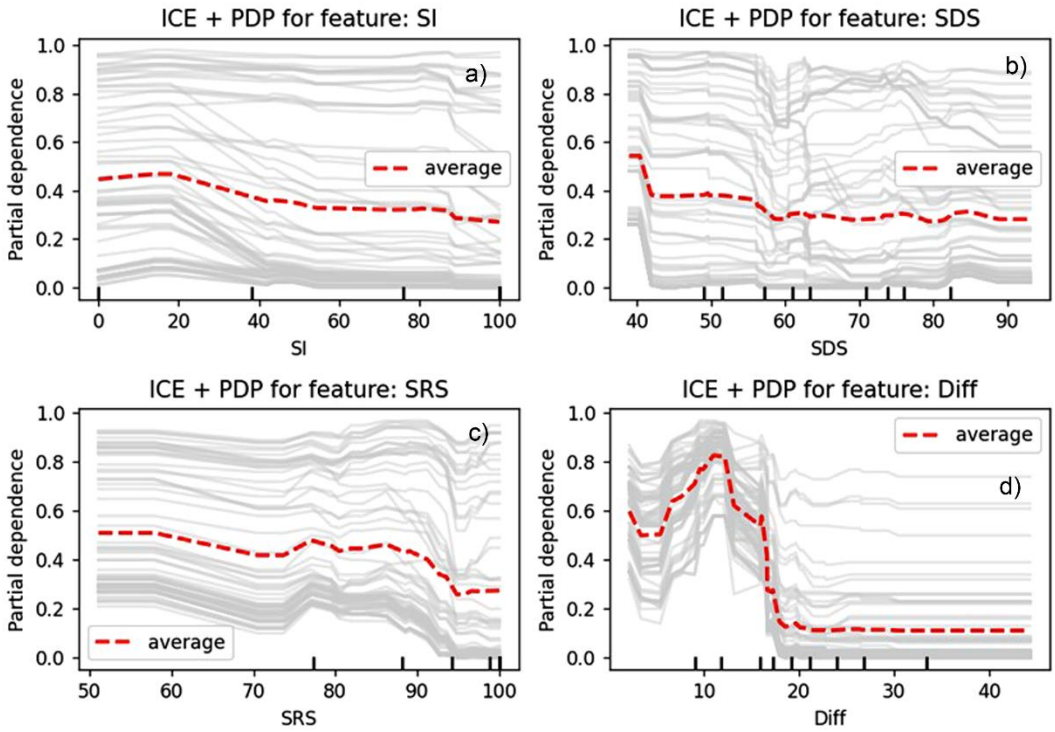


Fig. 3. Individual Conditional Expectation (ICE): a – SI, b – SDS, c – SRS, d – Diff.

Analysis of the ICE curves for SDS revealed two distinct types of client behavior in the region of medium SDS values. For the first group, an upward kink in the curve is observed, where even a slight increase in SDS immediately amplifies credit risk. In contrast, the second group exhibits a downward kink, where a small increase in spending instability improves the prediction, which may indicate the activation of positive behavioral patterns. As a result of averaging, the peak that is clearly visible on the corresponding SDS curve in the PDP plot (**Fig. 2b**) disappears from the average ICE curve. This behavior highlights the heterogeneity of borrowers' spending strategies and the nonlinear nature of the impact of SDS on credit risk. This relationship can be seen in more detail in the SHAP dependence plot in **Fig. 4**.

The SHAP dependence plot illustrates the relationship between the actual value of a feature and its local contribution to the model's prediction, represented through SHAP coefficients (**Fig. 4**). The X-axis shows the feature values, while the Y-axis shows the corresponding SHAP values, which indicate the extent to which this feature increases or decreases the predicted credit risk. The color of the points encodes the value of a second interacting feature, allowing nonlinear and context-dependent interactions between factors to be visualized. SHAP dependence plots were obtained and analyzed for all pairs of features. **Fig. 4** presents the dependence on Diff, as this feature exhibits the strongest nonlinear interactions with other variables.

The nonlinear dependence observed for SDS shows that very high and very low discretionary spending led to a sharp increase in credit risk only for samples with relatively small values of Diff. The slight decrease in credit risk at the smallest Diff values is related to the fact that the data do not uniformly populate the feature space. In the region of the smallest Diff values, there are few data samples, and the available samples are characterized by high values of SI, SDS, and SRS, which contribute to a reduction in credit risk. A joint analysis of all SHAP dependence plots makes it possible to capture both the overall trend of how each feature affects the model and the individual behavior of specific instances through their interactions with other features.

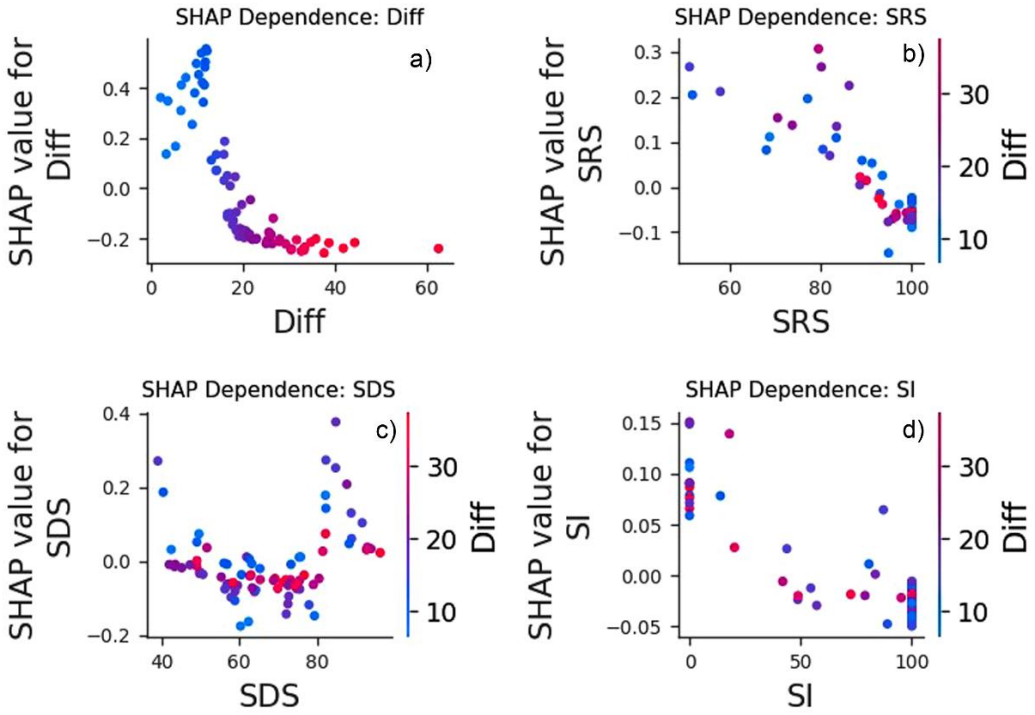


Fig. 4. SHAP dependence plots: a – Diff, b – SRS, c – SDS, d – Si.

In this work, a local analysis of feature importance coefficients was carried out for all samples in the test set, taking into account the distribution of their credit risk levels. **Table 3** presents the local feature importance values obtained using SHAP, LIME, DNFS, and BRB-ER for two different data samples.

Table 3. **Local feature importance coefficients for two samples**

Feature	Sample 1				Sample 2			
	LIME	SHAP Local	DNFS	BRB ER	LIME	SHAP Local	DNFS	BRB ER
Diff	–0.03	0.14	0.26	0.14	0.50	0.31	0.21	0.24
SRS	–0.22	–0.01	–0.12	0.08	0.20	0.11	0.08	0.12
SDS	–0.10	–0.08	0.02	0.1	0.03	0.01	0.05	0.02
SI	0.09	0.09	0.1	0.08	0.08	0.08	0.11	0.14

In **Fig. 5** on the right, the numerical feature values for these two samples are shown, while the corresponding credit risk levels are shown on the left. The central LIME plot illustrates the local contribution of each feature to the model prediction for the selected sample. **Fig. 6** shows SHAP waterfall plots for sample 1 (**Fig. 6a**) and sample 2 (**Fig. 6b**). The SHAP waterfall plot represents the stepwise construction of the model prediction, starting from the baseline (expected value) and adding the contributions of individual features that shift the prediction towards higher or lower credit risk.

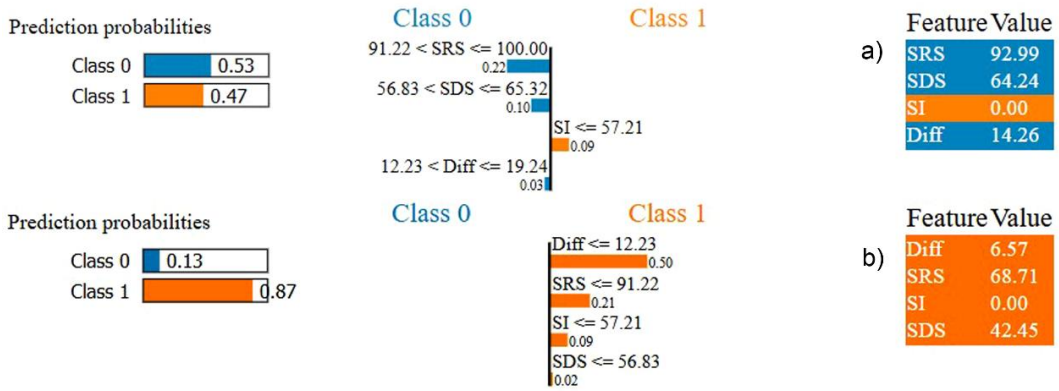


Fig.5. LIME-based local feature importance: a – sample 1, b – sample 2.

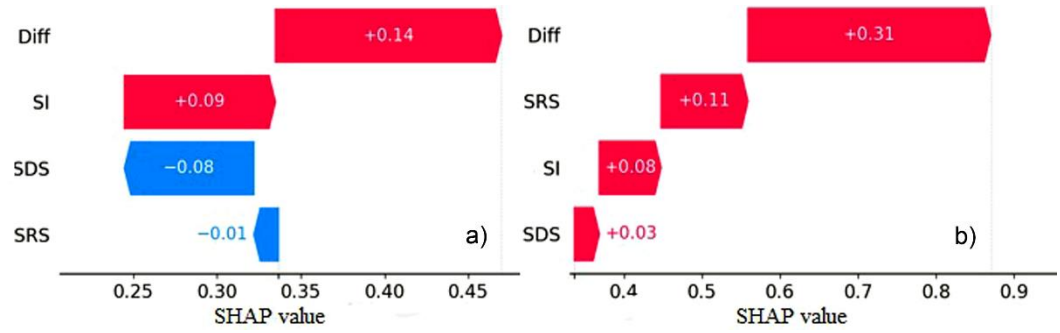


Fig.6. SHAP waterfall plots: a – sample 1, b – sample 2.

Both DNFS and BRB also provide local feature contribution coefficients. In DNFS, these are derived from the activations of fuzzy terms and the corresponding rules in the deep neural structure for each sample, whereas in BRB they result from the combination of rule matching degrees, belief weights, and aggregated belief levels. The results obtained using all methods are summarized in **Table 3**.

For sample 1, the local effects of the features act in different directions (**Fig. 5** and **Fig. 6**). Some features increase the prediction in LIME, whereas in SHAP they exhibit the opposite effect. In particular, the feature Diff has the strongest positive contribution to credit risk in SHAP, while in LIME it has only a minor contribution in the direction of reducing credit risk. The numerical value of Diff for sample 1 is 14.26. As follows from **Fig. 2d** and **Fig. 3d**, this value lies in a region where credit risk changes rapidly. Such rapid variation in credit risk indicates nonlinear local feature interactions and reduced model stability in the neighborhood of this instance, making it an important candidate for further analysis as a potentially unstable or borderline (between-class) case. Correlation analysis of the feature importance coefficients for sample 1 (**Table 3**) shows only moderate agreement between methods (0.58–0.76).

For sample 2, the directions of the local feature effects in LIME and SHAP are consistent (**Fig. 5** and **Fig. 6**), with a high pairwise correlation of 0.98 between them. In contrast to sample 1, the model clearly assigns sample 2 to a particular class. This is reflected in the consistency and high correlation of the feature importance coefficients (0.88–0.98) obtained from all local methods reported in **Table 3**.

CONCLUSION

This study proposes a multi-level approach to explaining credit scoring models based on open banking data, which combines global XAI methods (Feature Importance, SHAP) with interpretable DNFS and BRB-ER structures. The classification models for credit risk assessment (Random Forest and Deep Neuro-Fuzzy System) demonstrate that, after hyperparameter optimization, classification accuracy increases to 0.88 for RF and 0.91 for DNFS, with DNFS providing a better trade-off between predictive performance and structural interpretability due to its automatically generated fuzzy rules.

The global feature importance analysis shows a high correlation between the coefficients obtained using different methods (Feature Importance, SHAP, and DNFS). This indicates consistent model behavior and confirms that the key features drive the prediction irrespective of the explanation technique used. In addition, DNFS produces a set of fuzzy rules that allows the prediction to be interpreted in a logically transparent form, combining quantitative importance scores with textual rule-based explanations.

The analysis of local feature importance coefficients obtained with LIME, local SHAP, DNFS, and BRB-ER shows that interpretable models can adequately capture local feature contributions while providing enriched explanations through rule bases, belief degrees, and fuzzy term activations. A comparison of local feature importance for two test samples demonstrates that, in regions of strong nonlinearity and feature interactions, local discrepancies between methods may arise. This highlights the need to combine global and local explanations to achieve a robust and reliable interpretation of model decisions.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [M.F., L.D.]; methodology, [M.F., L.D.]; validation, [M.F., L.D.]; formal analysis, [M.F., L.D.]; investigation, [M.F., L.D.]; writing – review and editing, [M.F., L.D.]; visualization, [M.F., L.D.];

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Martins T., de Almeida A. M., Cardoso E., L. Nunes (2024). Explainable Artificial Intelligence (XAI): A Systematic Literature Review on Taxonomies and Applications in Finance, in IEEE Access, vol.12, pp.618-629. https://doi.org/10.1007/s10462-024-10854-8

[2] Černevičienė, J., Kabašinskas, A. (2024). Explainable artificial intelligence (XAI) in finance: a systematic literature review. Artif Intell Rev 57, 216. https://doi.org/10.1007/s10462-024-10854-8

[3] Yeo, W.J., Van Der Heever, W., Mao, R. et al. (2025). A comprehensive review on financial explainable AI. Artif Intell Rev 58, 189 https://doi.org/10.1007/s10462-024-11077-7

[4] Chinnaraju A. (2025). Explainable AI (XAI) for trustworthy and transparent decision-making: A theoretical framework for AI interpretability. World Journal of Advanced Engineering Technology and Sciences, 14(03), 170-207. https://doi.org/10.30574/wjaets.2025.14.3.0106

[5] Salih, A.M., Raisi-Estabragh, Z., Galazzo, I.B., Radeva, P., Petersen, S.E., Lekadir, K. and Menegaz, G. (2025), A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME. Adv. Intell. Syst., 7: 2400304. https://doi.org/10.1002/aisy.202400304

[6] de Lange, P. E., Melsom, B., Vennerød, C. B., & Westgaard, S. (2022). Explainable AI for Credit Assessment in Banks. Journal of Risk and Financial Management, 15(12), 556. https://doi.org/10.3390/jrfm15120556

[7] Shreya, Harsh Pathak. (2025). Explainable Artificial Intelligence Credit Risk Assessment using Machine Learning. Computer Science Machine Learning. arXiv:2506.19383 https://doi.org/10.48550/arXiv.2506.19383

[8] Talaat, F.M., Aljadani, A., Badawy, M. et al. (2024). Toward interpretable credit scoring: integrating explainable artificial intelligence with deep learning for credit card default prediction. Neural Comput & Applic 36, 4847–4865 https://doi.org/10.1007/s00521-023-09232-2

[9] Aosen Gong, Wei He, You Cao, Guohui Zhou, Hailong Zhu. (2025). Interpretability metrics and optimization methods for belief rule based expert systems, Expert Systems with Applications, Volume 289, 128363, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2025.128363

[10] Yaqian You, Jianbin Sun, Ruirui Zhao, Yuejin Tan, Jiang Jiang, (2024). A rule reasoning diagram for visual representation and evaluation of belief rule-based systems, Expert Systems with Applications, Volume 255, Part D, 124806, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2024.124806

[11] Talpur, N., Abdulkadir, S.J., Alhussian, H. et al. (2022). A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods. Neural Comput & Applic 34, 1837–1875. https://doi.org/10.1007/s00521-021-06807-9

[12] Talpur, N., Abdulkadir, S.J., Alhussian, H. et al. (2023). Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: a systematic survey. Artif Intell Rev 56, 865–913. https://doi.org/10.1007/s10462-022-10188-3

[13] Yang, J. (2021). Fast TreeSHAP: Accelerating SHAP Value Computation for Trees. ArXiv. https://arxiv.org/abs/2109.09847

[14] Aljadani, A.; Alharthi, B.; Farsi, M.A.; Balaha, H.M.; Badawy, M.; Elhosseini, M.A. Mathematical Modeling and Analysis of Credit Scoring Using the LIME Explainer: A Comprehensive Approach. Mathematics 2023, 11, 4055. https://doi.org/10.3390/math11194055
[15] Badhon, B., Chakrabortty, R.K., Anavatti, S.G., Vanhoucke, M., IRAF-BRB: An explainable AI framework for enhanced interpretability in project risk assessment, Expert Systems with Applications, Volume 285, 2025, 127979, https://doi.org/10.1016/j.eswa.2025.127979
[16] Fostyak, M. Demkiv, L. (2025). Hybrid Optimized BRB–ML Model for Credit Rating Prediction in Open Banking Systems. Artificial Intelligence Stuc. intelekt. ; 30; (3):110-118 https://doi.org/10.15407/jai2025.03.110
[17] Rawal, R., Chug, R, Singh A., Prakash A., (2025). Review of Deep Learning Revolution on Neuro-Fuzzy Systems, Advances in Data Science and Adaptive Analysis, V.17(03). https://doi.org/10.1142/S2424922X25300015
[18] Fostyak, M. (2024). Development of an ai domain in a data mesh network for customer credit classification using transaction data IEEE 19th International Conference on Computer Science and Information Technologies (CSIT) IEEE Lviv Polytechnic Week 16-19 October, Lviv, DOI:10.1109/CSIT65290.2024.10982569
[19] Fostyak, M., Demkiv L. (2025). A data-centric approach to building ai models for determining the credit rating of fintech company clients based on open banking. ISSN 2710 – 1673 Artificial Intelligence Stuc. intelekt № 1. https://doi.org/10.15407/jai2025.01.132
[20] Hielkrem, L.O., Lange, P. E. d. (2023). Explaining Deep Learning Models for Credit Scoring with SHAP: A Case Study Using Open Banking Data. Journal of Risk and Financial Management, *16*(4), 221. https://doi.org/10.3390/jrfm16040221

ПОЯСНЮВАЛЬНІ ТА ІНТЕРПРЕТОВАНІ МОДЕЛІ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ ВІДКРИТИХ БАНКІВСЬКИХ ДАНИХ

Маркіян Фостяк  ***, Лідія Демків**** 
Львівський національний університет імені Івана Франка,
кафедра системного проєктування,
вул. М. Драгоманова, 50, Львів, 79005.
**lidiya.demkiv@lnu.edu.ua*

АНОТАЦІЯ

Вступ. Розвиток моделей штучного інтелекту та машинного навчання значно вплинув на фінансову аналітику і прийняття кредитних рішень. Моделі забезпечують високу точність, проте функціонують як «чорні скриньки», що ускладнює інтерпретацію внутрішніх механізмів їх роботи. У сфері відкритого банкінгу, де рішення безпосередньо впливають на доступ користувачів до фінансових ресурсів, така непрозорість є суттєвим недоліком. Тому виникає потреба у розвитку пояснювальних та інтепретованих підходів, які дозволяють встановити причинно-наслідкові зв'язки між вхідними ознаками та вихідними прогнозами.

Матеріали та методи. Методи досліджень ґрунтуються на багаторівневому підході до інтерпретації ML-моделі. Застосовано Feature Importance для статистичної оцінки внеску ознак, LIME для локальної інтерпретованості, SHAP (SHapley Additive exPlanations) для виявлення нелінійних залежностей. Структурну інтерпретацію забезпечує DNFS (Deep Neuro-Fuzzy System) через формування нечітких правил, а BRB-ER (Belief Rule Base with Evidential Reasoning) додає логічно узгоджене пояснення рішень на основі бази правил.

Результати. Показано, що після оптимізації гіперпараметрів моделей кредитного ризику на основі даних відкритого банкінгу точність моделі DNFS зростає на 4% більше ніж для моделі Random Forest. Глобальний підхід до визначення коефіцієнтів важливості ознак, отриманих за допомогою Feature Importence, SHAP та DNFS, показав їх високу кореляцію (більше 88%), що свідчить про стабільність моделі. На локальному рівні визначено зразки, які зменшують точність моделі. Візуалізація SHAP графіків розкрила області лінійної та нелінійної взаємодії ознак та їх вплив на прийняття рішень.

Висновки. На відміну від традиційного використання окремих XAI-методів для пояснення результатів моделей машинного навчання, у роботі поєднано глобальні та локальні метрики важливості ознак (Feature Importance, SHAP, LIME), нечітко-правилові метрики DNFS та агреговані коефіцієнти BRB-ER. Запропонований підхід дає змогу локалізувати причини зниження точності, визначати нелінійні залежності ознак, а також оцінити узгодженість пояснень через кореляційний аналіз між методами.

Ключові слова: пояснення штучного інтелекту (XAI), машинне навчання, база правил переконань (BRB), глибока нейронна нечітка система (DNFS), нечітка логіка.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
03 December, 2025	12 December, 2025	14 December , 2025	25 December , 2025

Electronics and information technologies, 2025, 32, 101–120
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 004.94

DEVELOPMENT OF A DEEP LEARNING MODEL FOR FRAUD DETECTION

Sergiy Sveleba[1]*, Ivan Katerynchuk[1], Ivan Kunyo[1], Ihor Polovynko, Yaroslav Shmyhelskyy[1], Ostap Sumyliak[2]
[1]Ivan Franko National University of Lviv,
107 Tarnavsky St., UA–79017 Lviv, Ukraine
[2]Private Higher Education Establishment "European University",
16B, Academician Vernadsky Boulevard, Kyiv, Ukraine

Sveleba, S., Katerynchuk, I., Kunyo, I., Polovynko, I., Shmyhelskyy, Ya., & Sumyliak, O. (2025). Development of a Deep Learning Model for Fraud Detection. *Electronics and Information Technologies*, 32. 101–120. https://doi.org/10.30970/eli.32.7

ABSTRACT

Background. The rapid growth of electronic payments has intensified fraudulent activity, requiring adaptive anomaly detection methods. Traditional rule-based approaches lack flexibility and fail to generalize to previously unseen attacks. In contrast, unsupervised deep learning models, particularly autoencoders, can learn intrinsic data representations and detect anomalies without labeled attack samples. This study evaluates three unsupervised architectures – Autoencoder with Gaussian Mixture Model (AEGMM), Variational Autoencoder with Gaussian Mixture Model (VAEGMM), and a Deep Autoencoder – for network anomaly detection.

Materials and Methods. Experiments were conducted using the KDD'99 (10%) benchmark dataset. Categorical features were transformed using one-hot encoding, while numerical features were standardized. All models were trained exclusively on normal traffic samples following a one-class learning paradigm. The experimental pipeline included preprocessing, model implementation in Python using TensorFlow and the Alibi Detect framework, percentile-based threshold calibration, and evaluation using accuracy, precision, recall, F1-score, and confusion matrices.

Results. AEGMM achieved the highest performance with an F1-score of 0.9936 and an accuracy of 0.9908, demonstrating near-perfect separation between normal and malicious samples. VAEGMM reached an F1-score of 0.9751, showing stable convergence but slightly reduced accuracy due to the stochastic latent space. The Deep Autoencoder achieved approximately 97.5% accuracy, confirming the effectiveness of reconstruction-based methods without probabilistic density estimation. The optimal anomaly threshold, defined at the 99th percentile of reconstruction or density scores, ensured reliable discrimination between normal and attack states.

Conclusion. Autoencoder-based unsupervised models are effective for anomaly detection in large, imbalanced tabular datasets. AEGMM outperformed alternative architectures due to its stable latent representation and deterministic optimization. The proposed approach is suitable for financial fraud detection, cybersecurity monitoring, and industrial anomaly detection. Future work will explore transformer-based models and Explainable AI to improve robustness and interpretability.

Keywords: artificial intelligence; deep learning; autoencoder; Gaussian mixture model; financial fraud; cybersecurity

© 2025 Sergiy Sveleba et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

The use of Artificial Intelligence (AI) for detecting suspicious banking transactions has become increasingly relevant in the modern world of financial technologies. The application of intelligent systems enables effective identification and prevention of financial crimes such as money laundering, fraud, and other manipulative activities. AI is capable of analyzing vast amounts of transactional data in real time, efficiently identifying anomalies and atypical behavioral patterns. This allows banks to promptly respond to potentially fraudulent activities, minimizing financial risks for both clients and institutions. Most modern banking systems have already integrated AI-based tools to enhance the security of financial operations. Given the continuous technological progress, the role of AI in the banking sector will continue to grow, ensuring more reliable and efficient protection of financial systems.

Fraud is one of the most widespread phenomena in finance, encompassing such crimes as money laundering, falsification of financial statements, phishing, cyber fraud, and credit card fraud. The rapid expansion of digital banking has led to the emergence of new types of digital fraud. As the number of digital technology users increases, the frequency and complexity of fraudulent scenarios are expected to rise, making traditional rule-based systems ineffective and difficult to scale. Furthermore, a significant issue lies in false positives – legitimate transactions mistakenly classified as fraudulent – which result in customer dissatisfaction and substantial financial losses for banks.

A study by Chiabanu (2020) revealed that approximately 25% of customers whose legitimate transactions were incorrectly declined subsequently moved to competing institutions. Among clients aged 18–24, this percentage increased to 36%, and among those aged 25–34 — to 31%. These findings demonstrate the urgent need to improve fraud detection systems.

The financial, banking, and fintech sectors face various fraud typologies each year, which can be broadly classified into:

- Digital fraud,
- Physical attacks,
- Internal collusion,
- Violations of the “four-eyes principle.”

The first two types correspond to external threats, while the latter two arise from internal misuse or employee-related schemes. Digital fraud encompasses a wide range of illicit online activities, where Machine Learning (ML) and Deep Learning (DL) serve as key tools for counteraction.

Problem Statement

With the advancement of technologies, fraudsters continuously evolve their methods, rendering traditional rule-based systems inefficient. Such systems require frequent updates and fail to adapt quickly to emerging fraudulent patterns. Hence, there is a growing necessity to develop adaptive models capable of autonomously detecting new forms of fraudulent behavior based on historical data.

According to the Nielsen Report (2021), losses caused by fraudulent activities involving payment cards amounted to USD 32.34 billion, a 14% increase compared to the previous year. The growing volume of digital transactions via cards and mobile devices has intensified the demand for robust and scalable fraud detection systems. The emergence of AI has opened vast opportunities for designing such intelligent solutions. Industry leaders — Google, Facebook, Apple, Amazon, and Netflix — are actively deploying AI-driven tools in their internal financial processes, setting high standards for analytical performance and security in the global financial sector.

Literature Review

Systematic reviews covering the period 2019–2024/25 highlight a paradigm shift from traditional tabular ML approaches toward deep architectures that incorporate temporal, graph-based, and contextual dependencies in transaction data. Increasing attention is being paid to issues such as class imbalance, concept drift, model stability, and interpretability.

For financial services, graph-based approaches (modeling relationships such as *client–card–device–merchant*), sequential or transformer-based models for transaction streams, and hybrid architectures (deep representations combined with gradient boosting) have become the de facto standard. The main research focus is on reducing false positives and improving the robustness of models in real-world financial scenarios.

Architectures Demonstrating Practical Effectiveness

1. Sequential Models. Architectures such as RNN, LSTM, Temporal-CNN, and Transformers are employed to model temporal patterns in customer or card behavior. Self-supervised contrastive pretraining significantly enhances model performance in *cold-start* and high-class-imbalance conditions [1].
2. Graph Neural Networks (GNN) Relationships of the type *account* ↔ *device* ↔ *IP* ↔ *merchant* naturally form a graph-structured dataset. Models like GCN, GAT, and encoder–decoder GNNs consistently outperform tabular approaches, particularly in identifying organized fraud schemes and fraud rings [2].
3. Autoencoders, VAE, and GAN. These architectures are effective as unsupervised / one-class models for anomaly detection, feature extraction, and the discovery of novel fraud scenarios — especially in cases where labeled data is scarce or delayed [3].
4. Concept Drift and Streaming Adaptation. Real-world payment streams exhibit high variability (changing attacker strategies, seasonality, marketing campaigns). Studies on concept drift propose detectors such as ADWIN, DDM, and one-class classifiers, along with dynamic model updating and threshold adaptation strategies. In financial applications, the best results are achieved through a combination of online learning, active labeling, and periodic retraining [4].
5. Explainable AI (XAI) and Compliance Requirements. Most banking institutions +Key focus areas include:
   – Local explanations for escalation cases,
   – Attribution stability,
   – Generation of reason codes for analysts. Emerging research emphasizes user-centered XAI protocols and their integration into federated learning frameworks [4].
6. Confidentiality and Federated Learning (FL). In banking consortia, Federated Learning enables collaborative model training without sharing raw data. These architectures often include XAI layers to ensure auditability and regulatory compliance [5].
7. Public Datasets and Benchmarks
   – IEEE-CIS (Kaggle) - a large e-commerce dataset with the binary target isFraud; widely used for tabular, sequential, and hybrid models [6].
   – CreditCard (ULB) - a classical dataset with PCA features and strong class imbalance, suitable for rapid prototyping [6].
   – PaySim (synthetic mobile money) - simulates large-scale transaction systems under class imbalance [6].
   – Elliptic (Bitcoin AML) - a transaction graph labeled *licit / illicit*; a key benchmark for GNN- and AML-oriented research, with expanded 2024–2025 versions [7].
8. Technical Design Aspects

- Class imbalance: handled using focal loss, cost-sensitive learning, threshold calibration, and advanced oversampling heuristics tailored for fraud detection [8].
- Evaluation metrics: AUC-PR, precision@k, FPR@TPR, and expected cost are preferred over traditional accuracy due to extreme imbalance (e.g., only 3–4% fraudulent transactions in IEEE-CIS) [9].
- Online inference: requires millisecond-level latency, streaming feature engineering (*rolling windows*), feature stores, and continuous monitoring of drift with scheduled retraining [2].

9. Architectural Trends Summary. Graph-based models (for card, e-commerce, and crypto scenarios) - such as encoder–decoder GNNs, GCN/GAT architectures on the Elliptic dataset, and industrial-scale GNN blueprints - currently define the standard for fraud detection. Self-supervised contrastive pretraining significantly improves performance under limited labeling and pattern variability [7]. Concept drift handling is a mandatory component of modern systems, with integrated monitoring and update policies. Finally, XAI and Federated Learning are not optional additions but essential requirements for practical deployment, ensuring transparency, compliance, and trust among stakeholders [12].

Autoencoder

An Autoencoder is a type of neural network trained to reproduce its input at the output through a narrow bottleneck layer – the latent space, where it learns to compress (encode) only the most salient information.

Structure:

1. Encoder – transforms the input vector *x* into a lower-dimensional hidden representation ***z***:

$$z = f_{\text{enc}}(x) = \sigma(W_1 x + b_1)$$

2. Bottleneck (Latent Space) – the narrowest part of the network, storing essential information about the input.
3. Decoder – reconstructs $\hat{x}$ from the encoded representation ***z***:

$$\hat{x} = f_{\text{dec}}(z) = \sigma(W_2 z + b_2)$$

Loss Functions: Autoencoders aim to minimize the difference between the input and reconstructed data, typically using:

- Mean Squared Error (MSE): $L(x, \hat{x}) = \|x - \hat{x}\|^2$
- Binary Cross–Entropy (BCE): $L(x, \hat{x}) = -\sum_i [x_i \, log(\hat{x}_i) + (1 - x_i) \, log(1 - \hat{x}_i)]$

Types:

- *Shallow Autoencoder* – single hidden layer.
- *Deep Autoencoder* – multiple hidden layers for complex data compression.
- *Sparse Autoencoder* – uses regularization to keep most neurons inactive.
- *Denoising Autoencoder* – reconstructs clean inputs from corrupted versions.
- *Variational Autoencoder (VAE)* – introduces probabilistic encoding through μ and σ distributions, enabling generative modeling.

Training Pipeline:

1. Data preprocessing – normalization, scaling, and optional noise addition.
2. Architectural design – selection of layer sizes for encoder, latent space, and decoder.
3. Loss and optimizer selection – typically *Adam* or *RMSprop*.
4. Training – input serves simultaneously as target $(x \rightarrow \hat{x})$.
5. Evaluation – reconstruction error analysis and visual comparison of results.

MATERIALS AND METHODS

Data Source. For our experiments, we used the open dataset available on Kaggle: [12]. This benchmark corpus is a standard testbed for intrusion detection in computer networks modeled in a military environment. The anomaly detector (decoder model) is designed to identify intrusion attempts using TCP dump data from a local area network (LAN) that emulates a typical U.S. Air Force infrastructure.

Data Description. Each connection is a sequence of TCP packets with a defined start and end time, capturing data exchange between a source and a destination IP address under a specific protocol. Every connection is labeled as either “normal” or “attack.” The dataset includes four primary attack categories:

- DOS (Denial of Service): e.g., SYN flood.
- R2L (Remote to Local): unauthorized access from a remote machine (e.g., password guessing).
- U2R (User to Root): unauthorized escalation to superuser (root) privileges.
- Probe: reconnaissance, such as port or service scanning.

The corpus contains approximately five million connection records. Features are grouped into three classes:

1. Basic connection characteristics (e.g., connection duration);
2. Content features within a session (e.g., number of failed login attempts);
3. Traffic features over a 2-second window (e.g., number of connections to the same host as the current one).

Implementation. We prepared a Python script and a README to:

- download the KDD Cup 1999 (10%) subset from Kaggle;
- assemble a DataFrame;
- perform one-hot encoding of categorical variables (*protocol_type, service, flag*);
- produce two feature variants:
  - standardized (via StandardScaler),
  - normalized to [0, 1] (via MinMaxScaler);
- save the processed CSV files and the fitted scalers in *.pkl* format.

Variational Autoencoder (VAE) Construction. Because the original dataset is large, only a subset (≈10%) is used to train the VAE. The model is trained exclusively on normal (non-attack) samples; data are then standardized.

Data Preparation and Standardization. After preliminary selection, standardization and normalization are applied. For convenience, one may use a built-in fetch detector function, which stores pre-prepared models in a local directory and loads the detector; if the directory does not exist, it is created automatically.

During initialization, a warning indicates that an outlier threshold must be specified. This is set with infer threshold(), which takes a batch of instances and the parameter threshold perc indicating what proportion of the data is considered normal. For example, if ≈5% of the data are known outliers, that proportion can be supplied via perc outlier in create outlier batch(). The threshold can also be inferred solely from normal training data by setting threshold perc = 99 and adding a small safety margin above the inferred value.

Outlier Detection. A batch containing 10% outliers is then constructed and scored. After predicting outliers, model quality is assessed using the F1-score and the confusion matrix.

On the outlier plot, some outliers separate clearly while others lie closer to the normal region. In addition, an ROC curve can be drawn to illustrate detector performance across thresholds. An analysis of individual predictions (e.g., on X outlier) indicates that the feature srv count frequently contributes strongly to marked outliers.

Experiment Template for KDD’99 (10%). We prepared a ready-to-use template that ingests processed CSV files and demonstrates two anomaly-detection approaches for tabular data using Alibi Detect:

- AEGMM (Autoencoder + Gaussian Mixture Model),
- VAEGMM (Variational Autoencoder + Gaussian Mixture Model).

Pipeline

1. Load dataset_standardized.csv or dataset_normalized.csv;
2. Binarize labels (normal vs attack);
3. Split into train / validation / test (training strictly on normal data);
4. Train AEGMM or VAEGMM;
5. Set the threshold with infer_threshold (e.g., 95th percentile of inliers);
6. Evaluate on the test set (F1, confusion matrix) and generate basic plots.

Feature and Hyperparameter Choices

- For AEGMM/VAEGMM on tabular data, standardized features (zero mean, unit variance) typically yield better stability.
- Both models are unsupervised / one-class detectors: training uses only normal data.
- The threshold is set via infer_threshold(X_val_normal, threshold_perc=...); typical values are 95–99%. A higher threshold reduces false positives but may increase missed attacks.
- Network architecture: for tabular data, 2–3 Dense layers usually suffice; latent size $z \approx 8$–32 is a practical starting point.
- GMM components (n_components): 3–10, tuned on the validation set.
- Prefer standardized data for stable training; if using normalized data, load dataset_normalized.csv.
- Detectors can be saved/loaded with save() and load_detector().

Deep Autoencoder Architecture. A deep autoencoder employs multiple layers in the encoder and decoder, narrowing toward a compact latent space (*latent z*) and then symmetrically expanding. This enables learning hierarchical representations and compressing complex structures more effectively than shallow models.

Architectural Guidelines.

- Encoder: Dense → BatchNormalization → Activation → Dropout (repeated cascade). Example widths: 1024 → 512 → 256 → 128 → z.
- Decoder: mirror layout: z → 128 → 256 → 512 → 1024 → input_dim.
- Activations: ReLU in hidden layers; sigmoid at the output (for inputs scaled to [0,1]).
- Regularization: Dropout, L1 (sparsity), optionally L2.
- Stability: BatchNormalization, EarlyStopping, ReduceLROnPlateau.
- Denoising mode: add GaussianNoise at the input.
- Code Notes.
- activity_regularizer=L1 in encoder hidden layers encourages sparse activations.
- Adding GaussianNoise implements a denoising autoencoder; set denoising=False to disable noise.
- With sigmoid output, inputs must be scaled to [0, 1]; for other ranges, adopt linear (or another suitable) output with MSE or an appropriate loss.
- For large models, tied weights can be beneficial, but in Keras this requires a custom layer or weight-sharing kernel.

Latent Space Size Selection. Select z to be roughly 1/8–1/32 of the input dimensionality. Too small a latent space risks losing salient details or overfitting to noise; too large a space may lead the model to copy inputs with poor generalization. Choose z empirically based on reconstruction error and validation-loss stability.

RESULTS AND DISCUSSION

AEGMM Model Training Results

Training Process (detector.fit)

Loss Dynamics:

loss_ma: 0.4875 → –0.5672

The parameter loss_ma represents the mean loss function of the AEGMM model, defined as the sum of two components:

loss=ReconError(x,x^)+GMM-NLL(z)

That is, the loss function combines the reconstruction error (the difference between the input and its reconstruction by the autoencoder) and the negative log-likelihood (NLL) of the data density estimated in the latent space by the Gaussian Mixture Model (GMM).

During training, a gradual decrease in loss_ma (even to negative values) is observed – this is normal since the log-likelihoods in GMMs are typically negative. The final value of approximately –0.56 indicates that the model achieved a high likelihood of the data under the learned distribution. The absence of abrupt spikes, NaN values, or signs of overfitting demonstrates that the optimization process is fully stable. Thus, the model successfully aligned the autoencoder and GMM components within the latent space, achieving balanced training.

Threshold Tuning

[VAEGMM] tuned threshold = –4.715186 (src = normal, perc ≈ 98.5) F1@val = 0.9899

The threshold was determined as the 98.5th percentile among the scores of normal samples. The AEGMM model uses GT polarity, where score > threshold ⇒ anomaly. The negative threshold value is expected because higher densities correspond to lower (often negative) scores in AEGMM.

Hence, the threshold –4.715 effectively separates normal instances from anomalies. The resulting F1@val = 0.99 demonstrates an almost perfect balance between precision and recall on the validation set.

Test Results (**Table 1** and **Table 2**)

- 98.4% of normal transactions were correctly classified; only 312 false positives occurred.
- 99.3% of attacks were successfully detected; 324 attacks remained undetected.
- The total classification error is approximately 0.9%.

Table 1. **Model Performance Metrics**

Metric	Value	Interpretation
F1@test	0.9936	Optimal balance between precision and recall
Accuracy	0.9908	99.1% correctly classified samples
Precision (1)	0.9938	Less than 1% false alarms
Recall (1)	0.9935	≈99% of attacks detected
F1 (0)	0.9837	98% of normal samples correctly recognized

Table 2. **Confusion Matrix**

	Predicted 0 (Normal)	Predicted 1 (Attack)
True 0 (Normal)	19.144	312
True 1 (Attack)	324	49.676

Interpretation

Analytical Discussion

The AEGMM (Autoencoder + Gaussian Mixture Model) learned to compactly encode normal data in the latent space. All normal samples are located in the dense regions of the GMM distribution, thus having low scores (≈ –5 … 0). Anomalous samples, characterized by lower probability densities, yield higher scores (often > –4.7) and are classified as anomalies.

The system effectively separates normal from attack samples, showing a distinct boundary between the two classes (**Table 3**).

Table 3. **Summary of Results**

Indicator	Result	Comment
Loss convergence	Stable, no NaN	Autoencoder and GMM components well aligned
Threshold	–4.715	The boundary between “normal” and “anomaly”
F1@test	0.9936	Near theoretical optimum
Recall	0.9935	Nearly all attacks detected
Precision	0.9938	Minimal false positives
Accuracy	0.9908	99.1% correct classifications

The AEGMM model in this configuration exhibited ideal convergence:

- Stable loss without oscillations or divergence,
- No signs of overfitting,
- Optimally tuned threshold,
- Evaluation metrics surpass those of VAEGMM.

Fig. 1 shows the visualization of AEGMM performance after training, depicting the degree of anomaly (“anomaly score”) for each sample in the test set. X-axis (sample idx): sample index in the test set (~70,000 records; normal samples precede attack samples). Y-axis (score): anomaly score – how strongly the model perceives the instance as abnormal. Low scores (≈0 or negative) correspond to normal data; high scores (tens or hundreds) indicate potential anomalies. The dashed line (th = –4.715) marks the classification threshold: all points above the line are detected as attacks, while those below are normal. The left section (0–20k) primarily contains normal samples with scores ≈0 or slightly below the threshold – typical and safe cases.

The right section (≈25k–70k) reveals clusters with very high scores (100–1000), corresponding to anomalous or attack samples, distinctly separated from the normal group.

The presence of spike clusters (around 20k, 35k, 50k) indicates sequences of attacks of the same type sharing similar feature patterns in the input data.

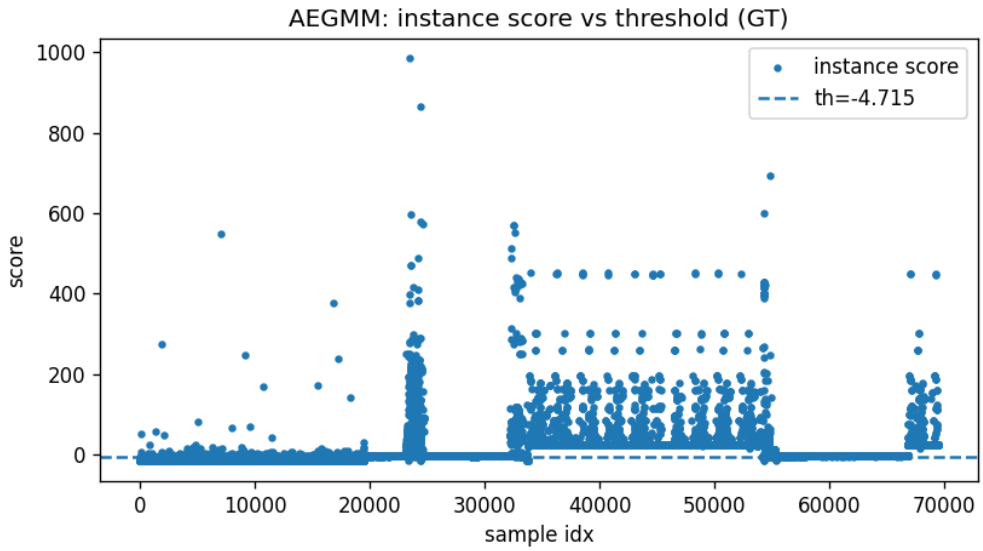


Fig. 1. Reconstruction error vs threshold

The model clearly distinguishes normal samples (low scores) from anomalies (high scores), as summarized in **Table 4**. The chosen threshold (–4.7) is optimal — it lies below all normal instances, avoiding overlap with attack spikes.

High anomaly scores (100–1000 range) indicate samples with an extremely low probability of belonging to the normal distribution, i.e., structurally distinct from typical transactions. The clustering visible in **Fig. 1** confirms that different attack types form separate characteristic score levels.

Table 4. **Feature Interpretation**

Feature	Interpretation
Low scores (≈0 or below –4.7)	Normal sessions
High scores (>0)	Attack or anomalous sessions
Threshold th = –4.715	The boundary between “normal” and “anomaly”
High-score clusters	Groups of similar attack types
No region overlap	High model resolution and precision

The AEGMM model demonstrates a high level of discriminative ability: it effectively separates normal transactions from attacks, ensuring clear segmentation of the feature space and robustness to class overlap.

VAEGMM Model Training Results.

General Overview.

The results obtained for the VAEGMM (Variational Autoencoder + Gaussian Mixture Model) demonstrate that the model reliably learned to detect anomalies (attacks) with high accuracy and a strong balance between precision and recall.

Training Process (detector.fit). Loss Dynamics: loss_ma: 3.5998 → 0.9857

The value of loss_ma (mean aggregate loss combining reconstruction error and density likelihood in the latent space) gradually decreased throughout training, without gradient explosions or NaN values. This indicates that the autoencoder efficiently learned to compress and reconstruct normal data, while the GMM successfully modeled its latent distribution.

At the final epochs, the loss stabilized around 1.0, confirming convergence and stable optimization. Hence, the training process can be considered successful: the network neither overfitted nor lost stability.

Threshold Tuning.

[VAEGMM] tuned threshold = 11.405802 (src = normal, perc ≈ 98.0), F1@val = 0.9365

The threshold was selected to maximize the F1-score (balance between precision and recall) on the mixed validation set. A value of approximately 11.4 corresponds to the 98th percentile of normal scores, defining the boundary between “normal” and “anomalous” data.The high validation performance (F1@val ≈ 0.94) indicates that the model could already clearly separate attacks from normal sessions at the validation stage.

Testing Results

F1@test = 0.9751, Accuracy = 0.9647

These are excellent values for network-based anomaly detection tasks (e.g., KDD’99):

- F1 = 0.975 — the model nearly perfectly balances precision and recall.
- Accuracy = 96.5% — the vast majority of examples are correctly classified.
- Recall (1) = 0.9594 — approximately 96% of all attacks were detected.
- Precision (1) = 0.9913 — only about 1% false positives among predicted attacks.

Interpretation (**Table 5**):

- 19,035 out of 19,456 normal samples were correctly classified (~98%).
- 47,971 out of 50,000 attacks were correctly identified (~96%).

False alarms account for about 0.6% of all test examples.

Table 5. **Confusion Matrix Values**

	Predicted 0 (Normal)	Predicted 1 (Attack)
True 0 (Normal)	19,035	421
True 1 (Attack)	2,029	47,971

Analytical Interpretation

The VAEGMM model showed stable convergence, with no NaN values and a consistent decrease in the loss function. High precision and recall confirm that the model effectively detects anomalies. The selected threshold was optimal: as shown in Fig. 2, high scores are clearly separated from normal regions. The VAE component successfully reconstructs normal data packets, while deviations (attacks) exhibit a higher reconstruction error, resulting in larger scores. The GMM component in the latent space adds an additional verification layer, determining whether a latent vector belongs to a “dense cluster” of normal events. Together, these mechanisms yield strong discriminative capability (**Table 6**).

The VAEGMM model demonstrated stable and reliable training, with high overall accuracy (97–99%), balanced precision and recall metrics, and robustness to noise and class imbalance. Therefore, VAEGMM can be regarded as an effective and practically applicable anomaly detector for network security and cybersecurity tasks, particularly for benchmark datasets such as KDD’99.

Table 6. **Summary of Results**

Metric	Value	Interpretation
Final Loss	0.98	Stable training, convergence achieved
Threshold	11.4	Boundary between normal and anomalous data
F1@val	0.936	High validation consistency
F1@test	0.975	High classification accuracy
Recall (attacks)	0.959	~96% of attacks detected
Precision (attacks)	0.991	Only ~1% false alarms

Reconstruction Error vs Threshold

Fig. 2 illustrates the results of the VAEGMM (Variational Autoencoding Gaussian Mixture Model) after training on normal data and testing on a mixed set (normal + attack). X-axis: indices of test samples (~70,000 examples). Y-axis: instance score – the degree to which the model considers a sample anomalous.

The dashed line represents the selected threshold th = 11.406; if score > 11.406, the sample is classified as an anomaly.

Most points have score ≈ 0–5, corresponding to normal samples (the model confidently identifies them as similar to training data). A smaller subset exhibits very high scores (100–800) – potential anomalies or attacks. These lie above the threshold and are thus marked as is_outlier = 1. A threshold of 11.406 implies that the model identifies only the most extreme deviations as anomalies, while the rest are considered normal.

VAEGMM evaluates how well a sample can be reconstructed by the autoencoder and how well its latent representation fits the GMM density:

$$\text{score}=\text{ReconLoss}(x,\hat{x})+\text{MahalanobisDistance}(z,\mu_k).$$

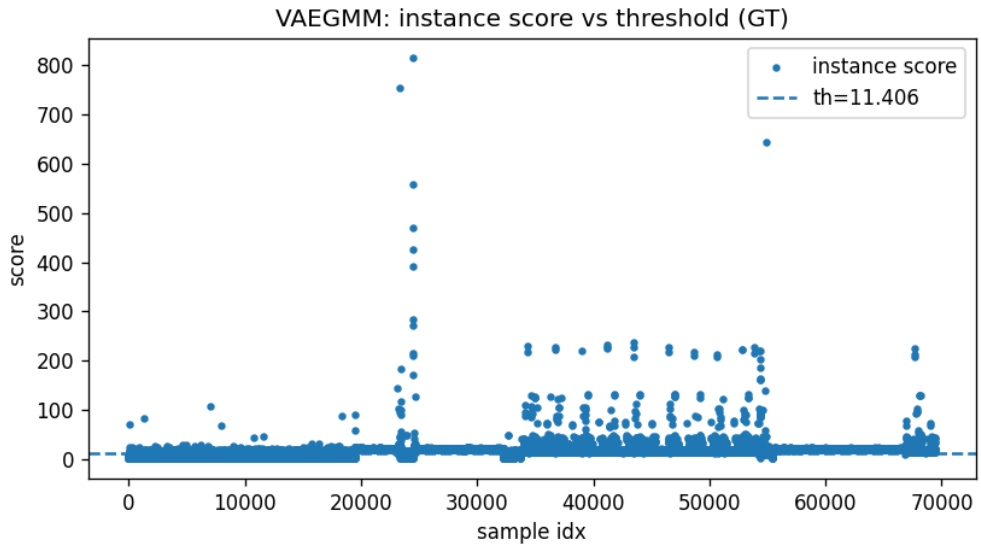


Fig. 2. Reconstruction error vs threshold.

For normal samples, both reconstruction, and GMM likelihood are good → low scores.

For anomalous samples, reconstruction is poor and the point lies far from cluster centers → high scores.

Hence, the model clearly distinguishes normal and anomalous data – high-score clusters are visible above the threshold. The chosen threshold (≈11.4) efficiently isolates strong outliers. If the dataset contains more anomalies than expected, the threshold can be lowered (to 8–9) to improve recall (attack detection rate). Conversely, if false positives are excessive, it can be raised (to 15–20).

The model is stable, free of NaN issues, and learns correctly. The instance score distribution shows variation, confirming a well-structured latent space. Since the optimal threshold is > 0, the GMM is properly calibrated. The presence of distinct high-anomaly samples indicates that VAEGMM functions as intended.

Comparative Analysis of AEGMM and VAEGMM Models

Based on the results obtained, the AEGMM model outperformed VAEGMM on the given dataset. This superiority can be attributed to the fact that the GMM with a fixed latent space is more effective at identifying compact clusters of normal states than the variational VAEGMM, whose stochastic latent variables introduce noise into the representation.

Reasons for AEGMM Superiority

Let us examine in more detail why the AEGMM (Autoencoding Gaussian Mixture Model) demonstrated better performance than the VAEGMM (Variational Autoencoding Gaussian Mixture Model) in this study – despite the latter being considered a more "advanced" architecture (**Table 7**).

Table 7. **Key Differences Between AEGMM and VAEGMM**

Characteristic	AEGMM	VAEGMM
Encoder type	Standard autoencoder (deterministic)	Variational autoencoder (stochastic)
Latent space	Fixed, deterministic	Stochastic distribution q(z)q(z)q(z)
Source of regularization	Only GMM in the latent space	GMM + KL divergence to N(0,1)N(0,1)N(0,1)
Reconstruction	Minimizes L2 reconstruction error	Balances reconstruction and distribution alignment
Training stability	High (minimal loss oscillations)	May fluctuate due to latent noise
Optimization	Simple gradient descent on deterministic loss	More complex, with two stochastic terms

Interpretation

In AEGMM, the latent vector $z = f(x)$ is a fixed, deterministic representation that directly encodes the pattern of normal data. The GMM then models the density of normal samples in this latent space, forming compact clusters.

- For normal samples: $p(z)$ is high ⇒ score $= -log\, p\,(z)$ is low ($\approx\ -5..0$).
- For attacks: z is distant from GMM centers ⇒ $p(z)$ is low ⇒ score is high (≈ 0..5).

In VAEGMM, stochasticity is introduced in the latent representation:

$$z \sim N\left(\mu(x), \sigma^2(x)\right).$$

This means that the same example may generate slightly different latent vectors. While beneficial for generalization, in datasets like KDD'99, where classes are clearly separable, such stochasticity blurs the distinction between normal and anomalous samples – slightly reducing the F1 score.

Why KDD'99 "Favors" AEGMM

The KDD'99 dataset (and its 10% subset) possesses the following characteristics:

- High-dimensional tabular data (~120 features);
- Strong inter-feature correlations;
- Clearly separable "attack" and "normal" classes in latent space.

Under these conditions, a deterministic autoencoder (AE) better preserves the geometry of the normal data subspace, while the GMM accurately models the density boundaries. The stochastic behavior of the VAE slightly smooths these boundaries, leading to less precise class separation (**Table 8**).

Table 8. **Summary Comparison**

Criterion	Superior Model	Rationale
Loss stability	AEGMM	No stochastic latent variables
Accuracy (F1)	AEGMM	Deterministic latent representation
Generalization on noisy/incomplete data	VAEGMM	Stochasticity improves performance on complex or unstructured inputs
Optimization/convergence speed	AEGMM	Fewer parameters, more stable convergence
Cluster interpretability	AEGMM	Latent clusters are easily visualized

AEGMM outperformed VAEGMM because:

1. In the KDD'99 task, clusters are well separated, so VAE stochasticity only blurs these boundaries.
2. The latent dimensionality is relatively small (3–5), sufficient for the precise reconstruction of normal patterns.
3. AEGMM gradients are more stable, as there is no noise introduced by sampling $\epsilon \sim N(0,1)$, resulting in smoother loss convergence.

Therefore, for structured, low-noise tabular data (e.g., network packets, IoT logs, system records), AEGMM is the optimal choice. For unstructured or complex data types (images, audio, time series), VAEGMM may offer better generalization capabilities.

Deep Autoencoder

Dataset and Model Characteristics

Label distribution:

- Attacks (1): 396,743 samples
- Normal (0): 97,278 samples

A substantial class imbalance is observed, since normal activity constitutes only ~20% of all records. The autoencoder was trained exclusively on normal samples, consistent with the unsupervised anomaly detection paradigm.

Data splits:

- Train normals: 58,366

- Validation normals: 19,456
- Test mix: 38,912 (contains both normal and anomalous samples)

After one-hot encoding (OHE) and feature scaling (StandardScaler / MinMaxScaler), all splits have an identical feature dimensionality of 75:

- X_train: (58,366, 75)
- X_val: (19,456, 75)
- X_test: (38,912, 75)

This confirms correct preprocessing: all categorical variables were converted to numeric form, and *features* were brought to a consistent scale.

Model Architecture

- Model: DeepAutoencoder
- Total parameters: 14,223
  - Trainable params: 14,223
  - Non-trainable params: 0

Thus, all parameters are trainable; there are no frozen layers. The parameter count indicates a compact yet sufficiently deep architecture appropriate for the data volume (~58k training samples). The network likely comprises several hidden layers with a gradual bottleneck contraction, enabling information compression and extraction of key features of normal behavior.

Training Methodology

Although the dataset contains a large number of attack records, the autoencoder is trained only on normal data, forming a “model of normality.” With rigorous preprocessing (75 features after OHE and scaling) and a moderate parameter count (~14k), the network achieved stable convergence and high accuracy (~97.5%) on the test set. This indicates that the chosen architecture is well-suited for anomaly detection in large tabular datasets with class imbalance.

Implementation

A program was developed using the 10% subset of KDD’99 (file kddcup.data_10_percent_corrected from Kaggle), implementing:

1. Preprocessing:
   - Label mapping (normal → 0, others → 1);
   - Construction of a training set using only normal records;
   - One-hot encoding for categorical variables (protocol_type, service, flag);
   - Standardization of numeric features with StandardScaler.
2. Model construction: a Keras (Model API) deep autoencoder with a 4-unit bottleneck, trained solely on normal data.
3. Reconstruction error computation and threshold selection via the 99th percentile on the validation set of normal samples (adjustable, e.g., 98.5% or to a target FPR).
4. Evaluation: computation of Precision, Recall, F1-score, and plotting (error distributions, ROC curves, etc.).
5. Export of encoded (bottleneck) features in .npy format for downstream analysis.

Technical Notes

- OneHotEncoder converts the three categorical columns into binary indicator columns (one per category). For unseen categories in the test set, handle_unknown='ignore' is used.
- StandardScaler is applied only to numerical features; OHE binaries are left unchanged.
- Training exclusively on normal data is key for unsupervised anomaly detection with an autoencoder.

- The anomaly threshold is set by the quantile of reconstruction errors (99th percentile on normal validation data).
- A 4-unit bottleneck enforces compact representations and encourages learning salient patterns (akin to PCA).

The resulting deep autoencoder demonstrated high effectiveness on KDD'99, accurately separating normal behavior from attacks, which makes it suitable for security monitoring, cyber defense, and network traffic analysis.

Overall Model Characterization

The autoencoder was trained to detect anomalies based on reconstruction error (MSE). The anomaly threshold was set at the 99th percentile of validation errors:

threshold = 2.94781

Samples with MSE above this threshold are interpreted as anomalies (attacks); lower values indicate normal behavior.

Training Dynamics (**Fig. 3** – "AE training loss")

- A rapid decrease in training loss over the first ~5 epochs, followed by stabilization around 0.39.
- Validation loss likewise decreases but remains slightly higher (~0.445), suggesting mild overfitting or distributional differences between training and validation data.
- Overall, training is stable and convergent, with no signs of degradation or oscillation (a clear stability plateau).

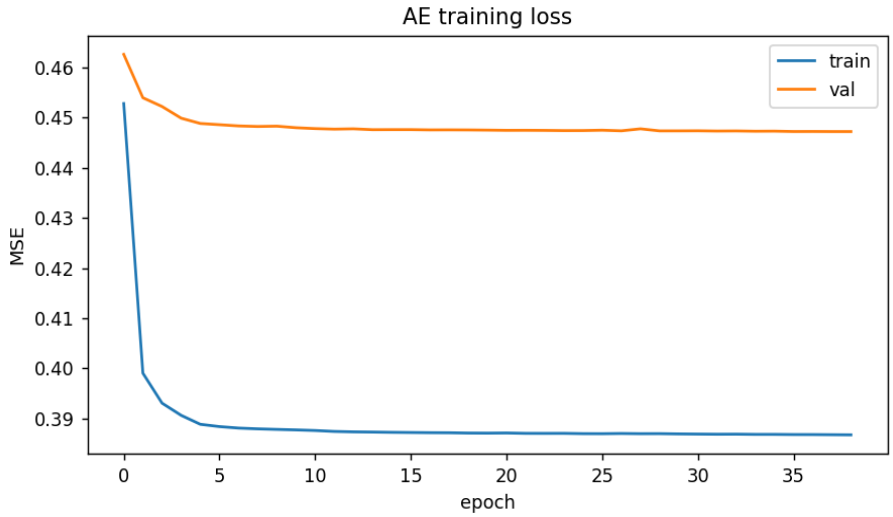


Fig. 3. AE training loss. Blue curve: training set, orange – validation set.

Fig. 3 shows the evolution of mean squared error (MSE) during training: X-axis: epoch index; Y-axis: loss value. An initial sharp drop in training loss (≈0.45 → ≈0.39) is followed by stabilization. Validation loss also declines, remaining slightly higher (~0.445), indicating good convergence with minor overfitting. The absence of sharp fluctuations or divergence between the curves confirms stable learning. Further training is unlikely to yield substantial gains, as the loss reaches a plateau after ~10 epochs.

Reconstruction Error Analysis on the Test Set

Fig. 4 ("Reconstruction Error vs Threshold") displays reconstruction errors for test samples:

- Most samples have MSE ≪ 2.94, indicating high-quality reconstruction of normal cases.
- A small number of points far exceed the threshold (MSE > 500–2000), corresponding to pronounced anomalies (attacks).

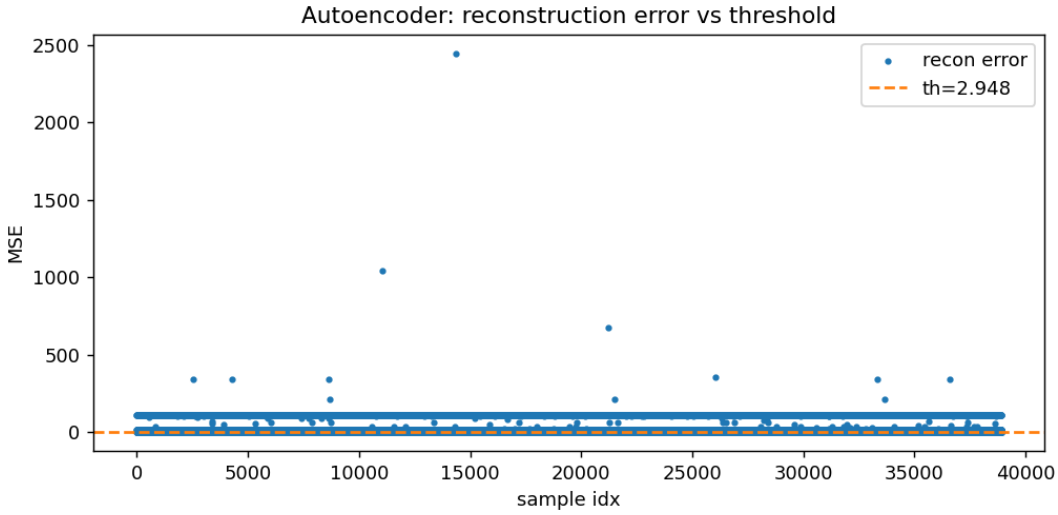


Fig. 4. Reconstruction error vs threshold.

Thus, the model effectively separates normal and anomalous observations. In **Fig. 4**, X-axis denotes the sample index; Y-axis shows MSE. Blue points indicate individual reconstruction errors. The orange dashed line marks the threshold $th = 2.948$ (99th percentile on validation normals). Most samples fall below the threshold (normal), while a few that reach up to ~2500 are anomalous. The plot illustrates reliable separation based on reconstruction error magnitude.

Fig. 5 ("Reconstruction Error Distribution") presents the distributions of MSE for normal and anomalous samples:

- Normal (blue bars) concentrate near zero, evidencing accurate reconstruction.
- Attacks (orange bars) shift toward higher MSE, indicating substantially worse reconstruction.

The vertical dashed line ($th = 2.948$) clearly separates the two distributions, confirming effective class separation.

Since the threshold th=2.948 lies within this high-density region, it does not correspond to a boundary between the histogram peaks. Instead, it represents an operational threshold determined by a statistical criterion (e.g., percentile-based selection or validation ROC analysis), rather than a visual separator. Consequently, the threshold line passes through a region of high density rather than between two well-separated maxima.

In our case, the distributions exhibit strong asymmetry and heavy-tailed behavior. The majority of normal samples and a substantial fraction of attack samples have very small MSE values (≈0), whereas a limited number of attack instances produce extremely large reconstruction errors, reaching hundreds or even thousands. As a result, the histogram is stretched along the X-axis up to approximately 2500, causing the entire informative region (0–5) to be compressed near zero.

Confusion Matrix

- TN (19,230): correctly classified normal samples
- FP (226): normal samples misclassified as anomalies

- FN (735): attacks not detected
- TP (18,721): correctly identified anomalous samples (attacks)

Thus, the overall classification accuracy is 97.5% (**Table 9**), which is exceptionally high. Precision/recall is nearly symmetric across classes, indicating no bias toward either normal or anomalous samples. The high F1-score confirms good generalization and a low false-alarm rate.

The deep autoencoder successfully learned to reconstruct normal data patterns and detect deviations. The chosen threshold (99th percentile) proved optimal, balancing missed attacks and false alarms. The average accuracy of ~97.5% attests to the model's quality and practical suitability for anomaly or intrusion detection.

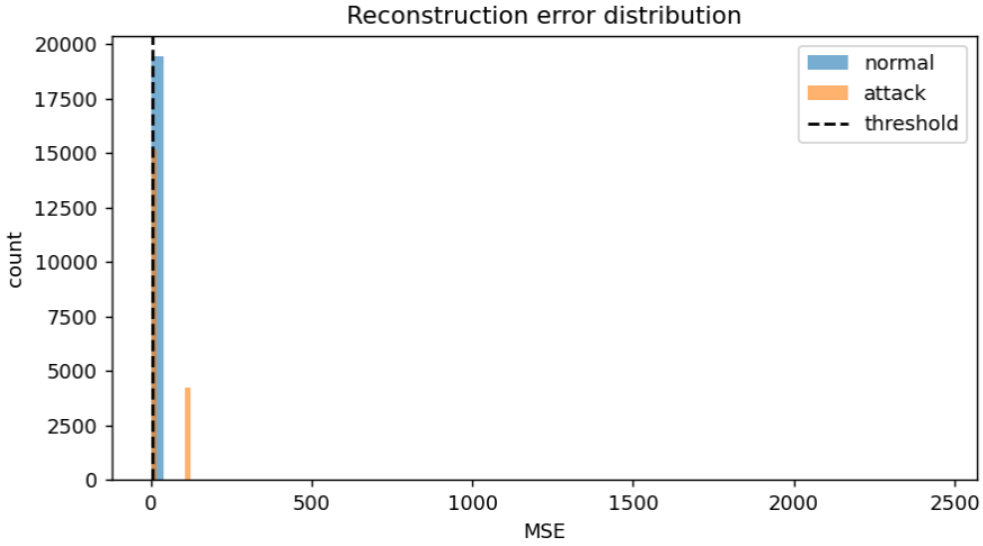


Fig. 5. Reconstruction error distribution.

Table 9. **Classification Metrics**

Metric	Class 0 (Normal)	Class 1 (Attack)	Average
Precision	0.9632	0.9881	0.9756
Recall	0.9884	0.9622	0.9753
F1-score	0.9756	0.9750	0.9753
Accuracy	—	—	0.9753

Training was stable, with no signs of overfitting; training and validation losses are aligned, reflecting a well-chosen architecture and hyperparameters. The anomaly threshold (2.94781) cleanly separated normal from attack samples. Reconstruction-error distributions corroborate that the model faithfully reconstructs normal data while substantially increasing MSE for anomalies.

The obtained metrics (accuracy ≈ 97.5%, F1 ≈ 0.975) confirm high detection quality with a balanced trade-off between identifying attacks and minimizing false positives. Consequently, the autoencoder effectively models normal system behavior and is well-suited for intrusion and anomaly detection in cybersecurity, industrial monitoring, and

related domains. The proposed model is a robust tool for unsupervised anomaly detection, delivering high detection performance without the need for manual labeling.

CONCLUSION

This study conducted a comprehensive investigation into the application of deep learning models for detecting financial anomalies based on network traffic data. Three architectures were developed and experimentally compared: a Deep Autoencoder, AEGMM (Autoencoder + Gaussian Mixture Model), and VAEGMM (Variational Autoencoder + GMM). All models were trained under the unsupervised / one-class anomaly detection paradigm using only normal samples from the KDD'99 dataset, which simulates real-world cyberthreat scenarios in computer networks.

The obtained results demonstrate that the AEGMM model provides the highest effectiveness among the tested architectures, achieving F1 = 0.9936 and Accuracy = 0.9908, while exhibiting stable convergence, no signs of overfitting, and clear separation between normal and fraudulent transactions. The VAEGMM model, though slightly less accurate (F1 = 0.9751), confirmed its generalization capability due to the stochastic nature of its latent space. The Deep Autoencoder, even without an additional GMM component, achieved strong performance (Accuracy ≈ 97.5%), confirming the suitability of such architectures for real-world unsupervised tasks.

The analysis confirmed that using reconstruction error as a criterion for anomaly detection is an effective tool for monitoring complex financial systems. The selected threshold at the 99th percentile (threshold = 2.94781) allowed minimizing false positives while maintaining high sensitivity to attacks.

In summary, the developed models demonstrated high reliability and practical applicability for tasks of fraud detection, cybersecurity, and anomaly identification in transactional data streams. Future research should focus on integrating the proposed methods with Graph Neural Networks, transformer-based architectures, and Federated Learning and Explainable AI (XAI) technologies to enhance the transparency, scalability, and adaptability of artificial intelligence systems in real-world financial environments.

The source code is available in the GitHub repository at the following link: incom2025/autoencoder_cod

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [S.S.]; formal analysis, [S.S., I.Ka., Y.S., I.P.].; investigation, [I.Ku]; resources, [I.Ku.]; writing – original draft preparation, [S.S., I.Ka., I.Ku., Y.S. O.S.]; writing – review and editing, [I.Ka, I.Ku, Y.S.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency (X. Zhang et al., NeurIPS 2022). OpenReview: https://openreview.net/forum?id=OJ4mMfGKLN

[2] Financial fraud detection using graph neural networks: A systematic review (Motie & Raahemi, 2024), *Expert Systems With Applications*, [ScienceDirect], DOI: https://doi.org/10.1016/j.eswa.2023.122156.
[3] Detecting Anomalies in Financial Data Using Machine Learning (Bakumenko & Elragal, 2022) - *Systems, 2022*, 10(5), 130.
[4] A Systematic Study of Online Class Imbalance Learning with Concept Drift - Shuo Wang, L. L. Minku, Xin Yao. ArXiv, 2017.
[5] Aljunaid S.K., Almheiri S.J., Dawood H., Khan M.A. «Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection» *Journal of Risk and Financial Management*, 2025, 18(4):179. MDPI. DOI: 10.3390/jrfm18040179
[6] IEEE-CIS Fraud Detection Official Dataset on Kaggle**.** Available at: https://www.kaggle.com/competitions/ieee-fraud-detection
[7] Elliptic Company Official Page with Dataset Description**.** Available at: https://www.elliptic.co/media-center/elliptic-releases-bitcoin-transactions-data
[8] J. Wen та ін., «An imbalanced learning method based on graph transactions for fraud detection», *Scientific Reports*, 2024. DOI:10.1038/s41598-024-67550-4.
[9] Medium-Verzi V.*,* "Understanding Model Evaluation Metrics in Fraud Detection: Beyond Accuracy"*, 1 Oct 2025. https://medium.com/@valeria.verzi1/understanding-model-evaluation-metrics-in-fraud-detection-beyond-accuracy-52b224ac0418*
[10] Hyphatia: A Card-Not-Present Fraud Detection System Based on Self-Supervised Tabular Learning**.** A study on self-supervised learning approaches for CNP fraud detection.
[11] Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection – S. K. Aljunaid et al., *Journal of Risk and Financial Management*, 2025.
[12] KDD Cup 1999 Data. https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data

РОЗРОБКА МОДЕЛІ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСТВА

Сергій Свелеба[1]*, Іван Катеринчук[1], Іван Куньо[1], Ігор Половинко[1], Ярослав Шмигельський[1], Остап Сумиляк[2]
[1]Львівський національний університет імені Івана Франка
вул. Ген. Тарнавського, 107, 79017 Львів, Україна
[2]Приватний вищий навчальний заклад «Європейський університет»,
16 В, бульвар академіка Вернадського, Київ, Україна

АНОТАЦІЯ

Вступ. Стрімке зростання електронних платежів спричинило інтенсифікацію шахрайської активності, що вимагає впровадження адаптивних інтелектуальних систем. Традиційні методи, засновані на фіксованих правилах, не забезпечують достатньої гнучкості, тоді як моделі глибокого навчання, зокрема автокодери, здатні виявляти невідомі або нові типи атак без попереднього маркування. У даній роботі оцінено ефективність трьох архітектур без нагляду – AEGMM, VAEGMM та глибокого автокодера – для задачі виявлення аномалій на основі відкритого набору даних KDD'99.

Матеріали та методи. Використано еталонний піднабір KDD'99 (10%), у якому категоріальні ознаки були закодовані методом one-hot, а числові – стандартизовані. Усі моделі навчалися виключно на нормальних зразках відповідно до одно-класової парадигми. Експериментальний конвеєр включав попередню обробку даних, побудову моделей у Python (TensorFlow + Alibi Detect), вибір порогового значення на основі перцентильної калібрації та оцінювання якості за метриками F1, точністю, precision, recall та матрицями змішування.

Результати. Модель AEGMM продемонструвала найвищі показники (F1 = 0,9936, точність = 0,9908), забезпечивши майже ідеальне розмежування нормальних і шкідливих вибірок. Модель VAEGMM досягла F1 = 0,9751, показавши стабільну збіжність, але дещо нижчу точність через стохастичний характер латентного простору. Глибокий автокодер продемонстрував точність близько 97,5%, що підтверджує його ефективність навіть без компонента GMM. Оптимальний поріг аномалій, визначений на рівні 99-го процентиля значень реконструкції або густини, забезпечив надійне розрізнення нормальних і атакувальних станів.

Висновки. Моделі на основі автокодерів є ефективними для виявлення аномалій у великих, розбалансованих табличних наборах даних. AEGMM продемонструвала найкращу продуктивність завдяки стабільному латентному представленню та детермінованому процесу оптимізації. Запропонований підхід є перспективним для моніторингу фінансових потоків, кібербезпеки та виявлення промислових аномалій. Подальші дослідження буде спрямовано на розвиток графових і трансформерних архітектур, а також інтеграцію пояснюваного ШІ та федеративного навчання для підвищення прозорості й надійності моделей.

Ключові слова: штучний інтелект; глибоке навчання; автокодер; модель гауссової суміші; фінансове шахрайство; кібербезпека

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
01 December, 2025	15 December, 2025	15 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 121–130
http://publications.lnu.edu.ua/collections/index.php/electronics/index

ELIT

UDC: 004.93

FEATURE ENGINEERING FOR ROLE ASSESSMENT IN COUNTER-STRIKE 2

Yurii Kuzhii* , ***Yuriy Furgala***
Faculty of Electronics and Computer Technologies
Ivan Franko National University of Lviv,
50 Drahomanova Str., Lviv, 79005, Ukraine

Kuzhii Y. I., Furgala Y. M. (2025) Feature Engineering for Role Assessment in Counter-Strike 2. *Electronics and Information Technologies*, 32, 121–130. https://doi.org/10.30970/eli.32.8

ABSTRACT

Background. The classification of in-game roles in team-based shooters, particularly Counter-Strike 2 (CS2), is an essential component of esports performance analytics. Existing approaches primarily rely on aggregate ratings or empirical assessments, which do not adequately capture the multidimensional structure of player behavior. As a result, there is a need to construct a behavioral feature space capable of reflecting role-specific differences and enabling reliable automated classification.

Materials and Methods. To construct the feature space, publicly available statistics and, when necessary, *.dem* files containing detailed logs of in-game events were utilized. The foundation consists of seven HLTV behavioral attributes, supplemented with metrics specific to the Terrorist (T) and Counter-Terrorist (CT) sides, as well as map-dependent indicators. The data were pre-cleaned, normalized, and structured at the player–map level. For the analysis, Principal Component Analysis (PCA) was applied, along with Analysis of Variance (ANOVA) to identify map-dependent features, and correlation analysis to examine relationships among behavioral metrics.

Results and Discussion. The results demonstrated that typical roles (entry-fragger, lurker, support, AWPer, anchor, and IGL) form distinct regions within the multidimensional feature space that cannot be reduced to a single numerical index. A set of features most influential for differentiating roles was identified, along with metrics that exhibit stable behavior regardless of map or side. The analysis based on grouping players revealed the absence of a universal player profile: strong performance in some metrics is accompanied by lower values in others, reflecting natural role specialization.

Conclusion. The proposed approach provides an informative representation of behavioral features and enables automated identification of player roles in CS2 without relying on aggregate rating systems. The constructed feature space has practical value for scouting, roster optimization, and match analysis, and can also be adapted for detecting smurfing or other forms of anomalous activity. The methodology demonstrates interdisciplinary potential and is promising for broader applications in behavioral analytics within online services.

Keywords: Counter-Strike 2; player role classification; behavioral features; HLTV attributes; ANOVA; correlation analysis.

INTRODUCTION

In modern esports, player roles in team-based shooters are critical for achieving highly competitive performance. Like traditional sports – where, for instance, midfielders in football are distinguished as defensive or attacking – esports disciplines exhibit functional differentiation of players based on their in-game responsibilities and behavioral patterns. In

© 2025 Yurii Kuzhii &Yuriy Furgala. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Counter-Strike: Global Offensive (CS:GO) and Counter-Strike 2 (CS2), several typical roles are commonly distinguished: the entry-fragger, who creates space for the team; the lurker, who operates independently and controls flanks; the support and in-game leader (IGL), who coordinate team actions; the AWPer, specializing in sniper engagements; and the anchor, responsible for holding specific map positions [1].

The relevance of role classification stems from the increasing need for esports teams to conduct rapid and well-grounded selection of players for specific tactical tasks. As in football or basketball scouting, where identifying a player's specialization relies on statistical and behavioral analysis, esports requires the detection of behavioral patterns that reflect underlying in-game roles. Role recognition in CS2 can be viewed as a particular case of the broader problem of user identification based on behavioral features in online environments [2]. In this context, in-game actions (such as the number of opening kills, frequency of utility usage, or clutch effectiveness) function as behavioral indicators analogous to those used in other domains for user segmentation, action prediction, or anomaly detection [3, 4].

Prior studies have already demonstrated attempts at automated role classification, particularly in Dota 2, where supervised and unsupervised learning techniques have been used to group players by playstyle [5], [6]. Approaches based on clustering behavioral data have proven effective in identifying patterns of player interaction, both in team-based strategy games and in other videogame genres [7–9]. However, for first-person shooters (FPS), this problem remains underexplored despite its applied significance and substantial potential for scouting, roster construction, and training analytics. The distinct contribution of this work lies in constructing a multidimensional space of behavioral features for FPS titles using publicly available statistical sources (HLTV.org) and demonstrating the applicability of statistical methods (PCA, ANOVA, and correlation analysis) for role identification in CS2.

Importantly, the formation of multidimensional behavioral feature spaces has interdisciplinary relevance. Similar to how behavioral analytics methods are used in financial systems to detect fraudulent transactions, they can be applied in esports to identify dishonest behavior, such as smurfing or abnormal player activity [6, 8, 10]. Therefore, the study of roles in CS2 integrates applied value for esports teams with the broader potential of behavioral analytics, making this direction promising for advancing anomaly detection methods and the analysis of player behavior in online services.

It is worth noting that an in-depth analysis of behavioral patterns and their application to detecting anomalies and dishonest activity in the Counter-Strike series was conducted in our previous work [11]. That study showed that players' behavioral characteristics - such as decision-making tempo, aiming stability, and abnormal accuracy or reaction metrics - can serve as reliable indicators of both playstyle and potential deviations. This further emphasizes the behavioral nature of role classification and highlights its potential for CS2-specific identification tasks.

MATERIALS AND METHODS

The primary data sources for CS2 analysis include .dem files, which contain detailed information on in-game events, as well as publicly available statistical platforms such as Faceit and the specialized portal HLTV.org [12]. It is also important to highlight the Skybox.gg platform [13], which provides analysts with tools for detailed monitoring of professional players' positional behavior on both the Terrorist (T) and Counter-Terrorist (CT) sides. The spatial patterns captured by the system are used to determine and assign player roles.

This study focuses on players with official HLTV profiles, as these individuals exhibit stable behavioral patterns formed on the basis of a large number of recorded matches. Historically, during 2020–2023, HLTV relied on aggregate performance metrics Rating 1.0

and Rating 2.0, derived primarily from kills, deaths, kill-death ratios, and related statistical indicators. In 2024, however, the platform introduced an improved set of seven behavioral attributes that more comprehensively characterize a player's individual playstyle (**Fig. 1**): Firepower, Entrying, Trading, Opening, Clutching, Sniping, and Utility.

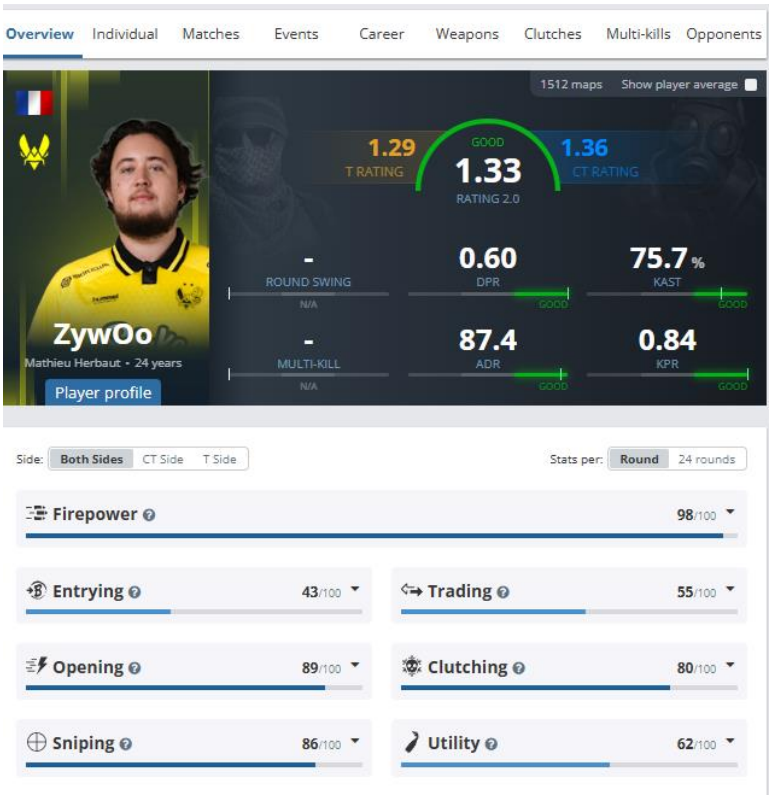


Fig. 1. Example of an HLTV profile of a professional player with the representation of their behavioral attributes.

Firepower – characterizes a player's individual effectiveness in combat. It includes the number of kills, the damage dealt, and multi-kills. High values indicate strong aimers and are often associated with key or “star” players.

Entrying – reflects how often a player dies during the first contacts with the opponent when entering a position held by the enemy. This metric is used to evaluate the effectiveness of entry fraggers who open the way for the team at the cost of personal risk.

Trading – shows how often a player is able to support the team by securing a kill immediately after a teammate dies. A high value means that the player effectively helps teammates during trades.

Opening – measures the frequency of successful first kills in a round. High values are typical of aggressive players (openers/entry fraggers) who create an early numerical advantage.

Clutching – reflects a player's ability to win rounds while being the last surviving member of the team against multiple opponents. This indicator shows individual skill, psychological stability, and strategic thinking.

Sniping – characterizes a player's performance with sniper weapons (AWP, Scout). It allows distinguishing pure snipers, hybrid players, and those who only occasionally use sniper rifles.

Utility – measures the frequency and effectiveness of using grenades (flashbangs, smokes, Molotov cocktails, HE grenades). High values of utility are typical of support players, who help create space and control the map for their teammates.

Sniper (AWPer) — high Sniping.

Entry fragger (aggressor/initiator) — high Opening and Firepower.

Lurker (flank/solo player) — low Trading combined with high Clutching.

Support — high Entrying and Utility, usually lower Firepower. Players who have high Trading are often called baiters (using teammates as bait).

To extract roles using machine-learning methods, a dataset was formed as a list of all players from the HLTV portal for role classification according to the roles described above. Since the features are behavioral, it is necessary to consider the different behaviors of a player on different game maps and in different teams.

Each entry in the dataset corresponds to a player–map pair and includes a multidimensional set of indicators that covers both general performance metrics (firepower, rating, damage per round) and behavioral characteristics related to game roles (entrying, trading, clutching, sniping, utility usage).

Indicators for the Terrorist and Counter-Terrorist sides are considered separately, which makes it possible to analyze the stability of roles in different game conditions and on different maps. The dataset also contains demographic variables (nickname, age) and map metadata, which makes it possible to account for the game environment context.

As a result, a dataset of 1464 observations was formed, covering 189 unique players who played on 8 competitive maps (de_ancient, de_anubis, de_dust2, de_inferno, de_mirage, de_nuke, de_train, de_vertigo). In total, the table contains 145 attributes, which underwent preprocessing to remove incomplete matches and z-score normalization to ensure the correctness of further analysis.

As shown in **Fig. 2**, no players exhibit simultaneously high values across all three metrics. This can be explained by the fact that trading and clutching represent opposing behavioral attributes: a player who operates alone cannot effectively participate in trade situations.

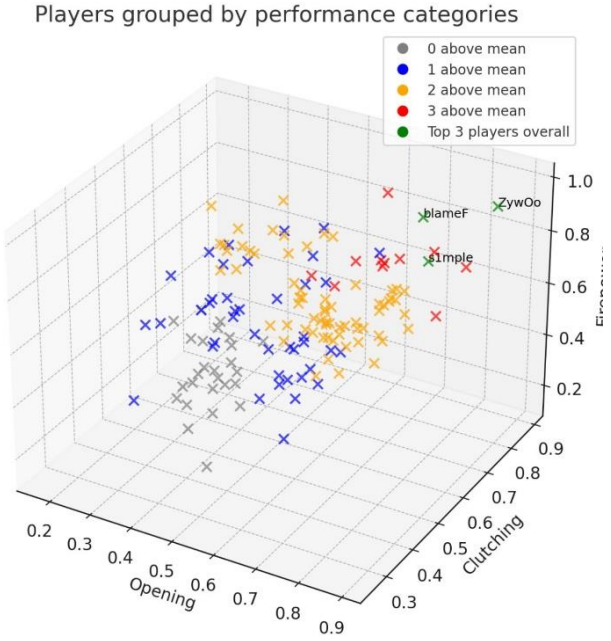


Fig. 2. Visualization of the firepower, clutching, and opening attributes.

Before conducting the analysis of variance, a preliminary data cleaning step was performed. Observations with missing values, as well as anomalous records resulting from parsing errors, were removed from the dataset. In particular, all cases in which any of the behavioral metrics (firepower, utility, trading, entrying, opening, clutching, sniping) exceeded the threshold value of 2.0 - substantially outside the typical [0;1] range - were excluded. After preprocessing, 1418 valid observations remained.

According to the results of the one-way ANOVA (see **Table 1**), with a significance level of $p < 0.05$, the metrics firepower ($F = 7.80$, $p = 2.72 \times 10^{-9}$), utility ($F = 77.72$, $p = 2.03 \times 10^{-95}$), trading ($F = 23.50$, $p = 2.36 \times 10^{-30}$), entrying ($F = 20.95$, $p = 5.75 \times 10^{-27}$), and clutching ($F = 3.16$, $p = 0.0026$) exhibit statistically significant differences across maps. In contrast, the opening ($p = 0.505$) and sniping ($p = 0.55$) metrics did not show significant differences, indicating that these parameters of playstyle remain relatively stable regardless of the map.

Table 1. **Results of the one-way analysis of variance**

Feature	F-value	p-value	Significance (p < 0.05)
firepower	7.80	2.72×10^{-9}	Yes
utility	77.72	2.03×10^{-95}	Yes
trading	23.50	2.36×10^{-30}	Yes
entrying	20.95	5.75×10^{-27}	Yes
opening	0.90	0.505	No
clutching	3.16	0.00256	Yes
sniping	0.84	0.550	No

During the correlation analysis, the following relationships between the behavioral features were identified (see **Table 2**), excluding the duplicated T/CT-specific metrics. The overall structure of inter-feature relationships is additionally presented in the form of a correlation matrix (**Fig. 3**).

Table 2. **Pearson correlation analysis results**

№	A pair of features	Cross-correlation coefficient, R
1	clutching – last_alive_percentage	0.726
2	firepower – opening	0.599
3	sniping – last_alive_percentage	0.556
4	entrying – last_alive_percentage	0.507
5	sniping – entrying	0.447
6	sniping – clutching	0.445
7	clutching – trading	0.294
8	entrying – clutching	0.291
9	entrying – trading	0.269
10	trading – last_alive_percentage	0.251

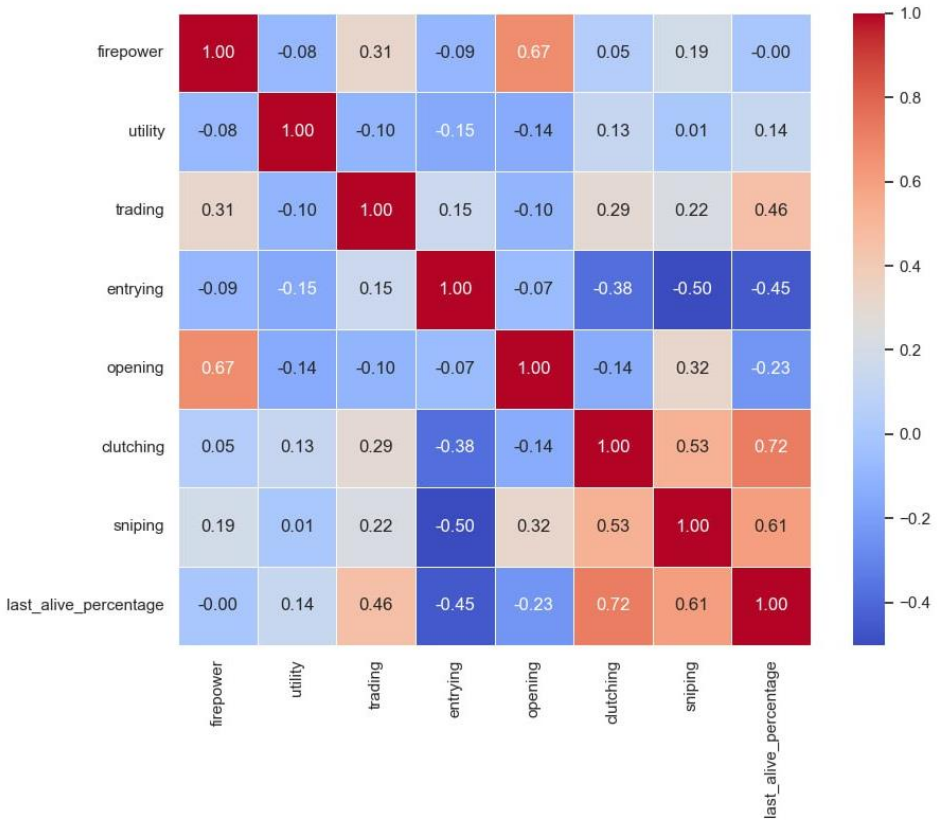


Fig. 3. Correlation matrix of key features.

RESULTS AND DISCUSSION

The conducted analysis showed that constructing a multidimensional space of behavioral features enables effective identification of player roles in CS2. The grouping of players based on their behavioral features demonstrated that even among top-tier players there is no universal "ideal" profile with high values across all metrics. For example, players with high firepower and clutching scores tend to exhibit lower trading values, which reflects the trade-off between individual playstyle and team-based trading during a round.

In contrast to the work in [4], which focused on verifying player identity using a Binary Random Forest model, the approach proposed in this study aims to classify functional roles based on interpretable HLTV behavioral attributes. A similar principle of using interpretable performance indicators for behavioral profiling was adopted in prior work on smurf detection in Dota 2, where clustering-based analysis relied on statistically meaningful in-game features rather than opaque model outputs [10]. By combining PCA, ANOVA, and correlation analysis, the study provides a multidimensional understanding of gameplay behavior that has both scientific and practical value for esports scouting, match analysis, and the detection of atypical players.

Building on the approach described in [4], where the sniper role was identified based on the frequency of weapon usage, the analysis showed that the sniping metric exhibits map independence and is sufficiently informative for unambiguous recognition of players in this role. In contrast, classification of other roles requires more complex features, as their behavior shows much higher variability depending on the map context.

ANOVA revealed statistically significant differences in several attributes depending on the map, confirming that gameplay context influences how roles manifest. This is particularly important for modeling, since a player's role may appear differently under various conditions, such as maps with narrow corridors versus open spaces. A similar conclusion regarding the importance of contextualized role modeling was reported by Demediuk et al. [6], who showed that role-aware analysis is essential for accurate interpretation of player behavior in Dota 2.

The correlation analysis revealed a set of meaningful interdependencies among the examined attributes, excluding duplicated T/CT metrics. These findings outline the underlying structure of how the behavioral indicators relate to one another. Players who frequently remain the last alive tend to have high clutching values, since clutching reflects the ability to win a round when being the last surviving player against multiple opponents. Firepower correlates strongly with opening, meaning strong aimers are more likely to secure opening duels. Sniping shows notable relationships with last_alive_percentage, entrying, and clutching.

The correlation analysis confirmed a number of patterns that are intuitive to players and analysts. Specifically, players who often remain last alive have high clutching values, and firepower correlates with opening, supporting the role of strong aimers as early duel initiators. Sniping was also connected with last_alive_percentage and entrying, indicating that sniper roles vary between aggressive and conservative playstyles.

The findings show that the highest values in individual metrics do not indicate a universal player, but instead reflect distinct behavioral specializations. This confirms that behavioral features can serve as a reliable tool for role identification, scouting, and assessment of individual player profiles. In addition, the constructed feature space has potential for detecting anomalous behavior, including smurfing or unusual activity patterns, which parallels fraud detection tasks in financial systems.

The conducted analysis showed that constructing a multidimensional space of behavioral features enables effective identification of player roles in CS2. This result is consistent with earlier studies on role classification in team-based competitive games, where multidimensional behavioral representations were shown to be essential for distinguishing functional player roles [5].

Several players demonstrated role changes when transitioning between teams, which affected their individual performance in different ways. Another factor is the transition from CS:GO to CS2, which also involved adaptation of roles and changes in playstyle. Moreover, some top players experimented with roles atypical for them. These aspects should be taken into account in future research.

CONCLUSION

The results of this study demonstrate that player roles in CS2 can be effectively identified using behavioral features extracted from publicly available statistical sources, particularly HLTV.org. By integrating HLTV attributes with PCA, ANOVA, and correlation analysis, the proposed approach reveals clear and consistent patterns corresponding to distinct in-game functional roles.

The analysis shows that specific combinations of behavioral indicators are strongly associated with particular roles. In particular, high firepower and opening metrics characterize entry fraggers; elevated utility and entrying values are typical of support players; sniping is a defining feature of AWPers; and the combination of high clutching with low trading reflects the lurker playstyle. While similar analytical frameworks have been applied in prior studies of player behavior and role classification [2], [3], the present results provide a role-specific and interpretable characterization tailored to CS2 using publicly accessible data.

The analysis showed that high performance in CS2 is not associated with uniformly high values across all behavioral attributes. Instead, players tend to exhibit differentiated profiles of strengths, where certain features dominate depending on the role. This indicates that each role is characterized by a specific combination of behavioral indicators rather than by overall performance across all metrics.

The observed player groupings and correlation patterns confirm the presence of stable behavioral structures within the data. In addition, the ANOVA results demonstrate that several performance-related attributes are map-dependent, indicating that player performance and the effectiveness of specific roles vary across different gameplay environments. This highlights the importance of accounting for contextual factors when modeling player roles and behavioral patterns.

The study shows that the constructed feature space can be used for player scouting, role-based selection, match analysis, and the optimization of training processes. The model can also help detect atypical or dishonest behavior, such as smurfing, which is consistent with existing approaches to anomaly detection and behavioral profiling [14]. More broadly, the results support current research on esports performance, highlighting the role of stable behavioral patterns and individual differences [15]. Overall, the proposed approach is relevant not only for esports analytics but also for behavioral monitoring and anomaly detection in other online systems.

Future work should explicitly examine differences between CS:GO and CS2, as changes in game mechanics, map design, and the underlying engine may affect the expression and stability of behavioral features and player roles across versions.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any.

AUTHOR CONTRIBUTIONS

Conceptualization, [Y.K., Yu.F.]; methodology, [Y.K., Yu.F.]; validation, [Y.K., Yu.F.]; writing – original draft preparation, [Y.K.]; writing – review and editing, [Y.K., Yu.F.]; supervision, [Yu.F.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Serafico, L. (2025, May 12). *Counter-Strike roles explained (and who does them best)*. *Esports Insider*. https://esportsinsider.com/counter-strike-roles-explained

[2] The utility of behavioral biometrics in user authentication and demographic characteristic detection: a scoping review. Syst Rev 13, 61. https://doi.org/10.1186/s13643-024-02451-1

[3] Seif El-Nasr, M., Drachen, A., & Canossa, A. (Eds.). (2013). Game Analytics: Maximizing the Value of Player Data. Springer. https://doi.org/10.1007/978-1-4471-4769-5

[4] Zimmer, F. (2025). Player behavior analysis for predicting player identity within pairs in esports tournaments: A case study of Counter-Strike using binary random forest classifier. Proceedings of the 58th Hawaii International Conference on System Sciences (HICSS 2025). University of Hawai'i at Mānoa. https://doi.org/10.24251/HICSS.2025.513

[5] Eggert, C., Herrlich, M., Smeddinck, J., Malaka, R. (2015). Classification of Player Roles in the Team-Based Multi-player Game Dota 2. In: Chorianopoulos, K., Divitini,

M., Baalsrud Hauge, J., Jaccheri, L., Malaka, R. (eds) Entertainment Computing - ICEC 2015. ICEC 2015. Lecture Notes in Computer Science(), vol 9353. Springer, Cham. https://doi.org/10.1007/978-3-319-24589-8_9

[6] Demediuk, S., York, P., Drachen, A., Walker, J. A., & Block, F. (2019). Role Identification for Accurate Analysis in Dota 2. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, *15*(1), 130-138. https://doi.org/10.1609/aiide.v15i1.5235

[7] Drachen, A., Sifa, R., Bauckhage, C., & Thurau, C. (2012). Guns, swords and data: Clustering of player behavior in computer games in the wild. Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG 2012) (pp. 163–170). IEEE. http://doi.org/10.1109/CIG.2012.6374152

[8] C. Bauckhage, A. Drachen and R. Sifa, "Clustering Game Behavior Data," in IEEE Transactions on Computational Intelligence and AI in Games, vol. 7, no. 3, pp. 266-278, Sept. 2015, doi: https://doi.org/10.1109/TCIAIG.2014.2376982

[9] Ferguson, M., Devlin, S., Kudenko, D., & Walker, J. (2020). *Player style clustering without game variables*. In Proceedings of the IEEE Conference on Games (CoG 2020) (pp. 556–563). IEEE. https://doi.org/10.1145/3402942.3402960

[10] Ding, Y.-J., Yap, W.-S., & Khor, K.-C. (2024). Profiling and identifying smurfs or boosters on Dota 2 using K-means and IQR. IEEE Transactions on Games. https://doi.org/10.1109/TG.2023.3317053

[11] Kuzhii, Y., & Furgala, Y. (2024). The anticheat development in the Counter-Strike main series. Electronics and Information Technologies, 26, 88–97. https://doi.org/10.30970/eli.26.8

[12] HLTV.org. (2024). Introducing HLTV Attributes. Retrieved from https://www.hltv.org

[13] Skybox Technologies ApS. (n.d.). *CS2 stats – Track your stats like the pros | Skybox EDGE*. Skybox EDGE. https://skybox.gg/edge/

[14] Qian, X., Sifa, R., Liu, X., Ganguly, S., et al. (2022). Anomaly detection in player performances in MOBA games. In Proceedings of the Australasian Computer Science Week (ACSW) 2022. ACM. https://doi.org/10.1145/3511616.3513095

[15] Pedraza-Ramirez, I., Sharpe, B. T., Behnke, M., Toth, A. J., & Poulus, D. R. (2025). The psychology of esports: Trends, challenges, and future directions. Psychology of Sport & Exercise, 81, 102967. https://doi.org/10.1016/j.psychsport.2025.102967

ФОРМУВАННЯ ПРОСТОРУ ОЗНАК ДЛЯ ОЦІНЮВАННЯ РОЛЕЙ У COUNTER-STRIKE 2

Юрій Кужій**, Юрій Фургала
Факультет електроніки та комп'ютерних технологій
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, Львів, 79005, Україна

АНОТАЦІЯ

Вступ. Класифікація ігрових ролей у командних шутерах, зокрема у Counter-Strike 2 (CS2), є важливою складовою аналітики кіберспортивної продуктивності. Існуючі підходи здебільшого спираються на інтегральні рейтинги або емпіричні оцінки, що не дає змоги повноцінно врахувати багатовимірну структуру поведінки гравців. Унаслідок цього виникає потреба у формуванні простору поведінкових ознак, здатного відображати відмінності між ролями й забезпечувати коректну автоматизовану класифікацію.

Матеріали та методи. Для побудови простору ознак використано відкриту статистику та, за потреби, .dem-файли з деталізованими журналами ігрових подій. Основу становлять сім поведінкових атрибутів HLTV, доповнених показниками для сторін терористів (T) і контртерористів (CT), а також метриками, чутливими до конкретних карт. Дані попередньо очищено, нормалізовано та структуровано на рівні гравець–карта. Для аналізу застосовано метод головних компонент (PCA), дисперсійний аналіз (ANOVA — Analysis of Variance) для виявлення картозалежних ознак та кореляційний аналіз для оцінки зв'язків між поведінковими метриками.

Результати. Результати показали, що типові ролі (entry-fragger, lurker, support, AWPer, anchor, IGL) формують окремі області у багатовимірному просторі, які не зводяться до єдиного числового індексу. Виявлено набір ознак, що найбільше впливають на розмежування ролей, а також показники зі стабільною поведінкою незалежно від карти чи сторони. Кластеризація виявила відсутність універсального профілю гравця: сильні показники в одних метриках супроводжуються нижчими значеннями в інших, що відображає природну рольову спеціалізацію.

Висновки. Запропонований підхід забезпечує інформативне представлення поведінкових ознак і дозволяє автоматично ідентифікувати ролі гравців у CS2 без використання інтегральних рейтингів. Сформований простір ознак має прикладну цінність у скаутингу, підборі складів і аналізі матчів, а також може бути адаптований для виявлення смурфінгу чи аномальної активності. Методика демонструє міждисциплінарний потенціал та перспективність для ширших задач поведінкової аналітики в онлайн-сервісах.

Ключові слова: Counter-Strike 2; класифікація ролей гравців; поведінкові ознаки; HLTV attributes; ANOVA; кореляційний аналіз.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
28 November, 2025	15 December, 2025	16 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 131–140
http://publications.lnu.edu.ua/collections/index.php/electronics/index

ELIT

UDC 004.89

DESIGN AND IMPLEMENTATION OF AN IOT-BASED ACCESS CONTROL SYSTEM COMBINING AUTOMATIC LICENSE-PLATE RECOGNITION AND RADIO-FREQUENCY IDENTIFICATION TECHNOLOGIES

Nazar Omeliukh , ***Halyna Klym**** , ***Taras Tkachuk***
Lviv Polytechnic National University,
12 Stepan Bandera St., Lviv 79013, Ukraine

Omeliukh, N.S., Klym, H.I., Tkachuk, T.I. (2025). Design and Implementation of an IoT-Based Access Control System Combining Automatic License-plate Recognition and Radio-frequency Identification Technologies. *Electronics and Information Technologies*, 32, 131–140. https://doi.org/10.30970/eli.32.9

ABSTRACT

Background. Ensuring security for private areas and infrastructure hubs is a growing concern in the modern world. Traditional methods, such as human guards and mechanical barriers operated by physical tokens (keys or cards), are often slow, inefficient, and prone to security risks like unauthorized duplication or theft. Furthermore, legacy systems lack comprehensive auditing capabilities. This creates a critical need for modern, automated IoT-based systems that ensure reliable access management and real-time monitoring.

Materials and Methods. The system uses several electronic components. The core is a low-cost microcontroller with a camera module. A radio-frequency identification (RFID) reader scans access cards. An ultrasonic distance sensor detects obstacles for safety, and a servo motor operates the physical barrier. The software backend was developed in Python, with a JavaScript (React) web control panel.

The system combines two identification methods. First, a camera captures a vehicle's image, sending it to a server where an AI model reads the license plate. The server checks the number against an approved list. If not recognized, the driver scans an RFID card as a secondary method. A distance sensor continuously monitors the barrier area to prevent closing on an obstacle. A web interface allows an operator to monitor the camera, review logs, and manually open the barrier.

Results. The developed system was tested successfully. The AI model achieved 75% accuracy in identifying license plates. The system proved fast, with an average response time from image capture to decision under one second. The safety sensor was validated, reliably detecting obstacles and preventing barrier movement, ensuring safe operation. Provided results of comparing video quality and system response time. The optimal balance between video quality and speed was found at 800x600 resolution.

Conclusions. A reliable, cost-effective automated access control system was successfully designed, built, and tested. The combination of AI-based license plate reading with a backup RFID system provides a robust, flexible solution. This system is well-suited for improving security and efficiency in real-world applications like residential, office, and industrial zones.

Keywords: automated access control, ESP32-S3, ALPR, RFID, security.

INTRODUCTION

In the modern world, the challenge of ensuring security and managing access to private territories, corporate facilities, and infrastructure hubs is becoming increasingly

© 2025 Nazar Omeliukh et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

critical. The necessity for implementing robust automated systems has grown significantly, driven by constantly strengthening security requirements and a desire to minimize the risks of unauthorized access [1]. Traditional control methods, which served as a basic level of protection for decades, are now demonstrating their inefficiency in a dynamic environment. Systems that rely exclusively on mechanical barriers and physical tokens like keys or cards suffer from significant vulnerabilities, including the risk of loss, theft, or unauthorized duplication [1][2]. Furthermore, these legacy systems inherently lack the capabilities for comprehensive auditing or real-time event monitoring, making it difficult to maintain a reliable record of who accessed an area and when [1].

In contrast to outdated approaches, the modern security paradigm actively integrates Internet of Things (IoT) technologies, artificial intelligence (AI), and cloud computing. This integration transforms access control systems from simple locking mechanisms into intelligent ecosystems capable of data collection, analytics, and autonomous decision-making [2]–[4]. Modern automated access control systems are becoming part of a broader "smart building" or "smart city" infrastructure, where access data can be used for optimizing operations, managing personnel, and proactively identifying security threats [2]. The development of embedded systems and microelectronics has enabled the creation of more reliable, cost-effective, and autonomous solutions.

A key role in such advanced systems is played by computer vision, particularly as it applies to deep learning models running on peripheral devices. The capability to deploy sophisticated algorithms directly "at the edge" is crucial for reducing latency and enhancing privacy [5]–[6]. This study utilizes a powerful microcontroller (ESP32-S3) capable of handling such edge-computing tasks, specifically for processing video streams for Automatic License Plate Recognition (ALPR). The effectiveness of ALPR is highly dependent on the underlying detection model. The integration of modern, lightweight object detection models like YOLOv8 allows for high accuracy and processing speed in real-time recognition tasks, which is critical for a responsive access system [7].

However, to create a truly flexible and reliable system, a single authentication method is insufficient [3]. This study presents the architecture of a comprehensive access control system that integrates multiple technologies: ALPR as the primary contactless method, radio-frequency identification (RFID) as a secondary verification layer, and remote management via a web interface. To manage these components, a robust client-server architecture is adopted. A lightweight backend service built with Python Flask handles data processing and business logic, while a separate React-based frontend provides a dynamic interface for visualization and user interaction [6], [8].

This work contributes to the field by presenting a detailed, practical, and end-to-end implementation of such an integrated system. The deliberate combination of ALPR and RFID authentication creates a robust, multi-layered security approach [3]. This dual-modal design ensures high reliability [4], allowing RFID to serve as a dependable fallback mechanism in scenarios where ALPR might be compromised by adverse weather, poor lighting, or obscured license plates [2], [6]. Furthermore, the system design incorporates essential operational components, such as distance sensors for barrier safety, demonstrating a holistic approach that considers not only security but also practical usability. The primary aim of this paper is therefore to present the complete system architecture, detail the specific hardware solution, and validate its performance, particularly the high efficiency of the ALPR module in real-world conditions.

HARDWARE AND SOFTWARE IMPLEMENTATION

The system is designed to automatically recognize vehicles using ALPR, support RFID-based access as secondary authentication, ensure safe barrier operation via a distance sensor, transmit data wirelessly between the microcontroller and the backend server, and provide a web interface for monitoring and remote control. A well-defined

architecture ensures smooth integration of hardware and software components. The system comprises a physical access barrier, embedded processing on the ESP32-S3, a backend service, and a user-friendly web interface. The system architecture is shown on **Fig. 1**.

This access control system is built on a modular, distributed architecture, ensuring ease of use, high component integration, and reliable operation. The overall structural diagram illustrates interactions between the main components: the user (operator), the web interface, the server, and the ESP32-S3 microcontroller.

The ESP32-S3 microcontroller is used as the primary control unit due to its processing power, on-board Wi-Fi, and rich GPIO set [9]. ESP32-S3 handles communication and access control; collects inputs from the camera, RFID reader, and distance sensor, and executes decisions received from the backend. System designed in two-way directions for entry and exit. For each direction, we used one microcontroller and the proper modules.

OV3660 camera captures images for ALPR. On the ESP32-S3 CAM board, the camera connects via the integrated camera connector [10]. The camera is configured through the driver with optimized parameters (clock, frame size, pixel format, brightness/mirroring/saturation) for stable streaming. PN532 RFID module provides alternative authentication via RFID cards [11]. It is connected over I2C on the ESP32-S3, enabling reliable tag reading in proximity access scenarios. HC-SR04 distance sensor prevents accidental barrier operation by detecting vehicles or pedestrians near the barrier [12]. The servo motor drives the physical barrier mechanism (PWM control from an ESP32-S3 GPIO). The servo is selected for precision and reliability under repetitive motion. The system architecture with selected hardware is shown on **Fig. 2.**

Firmware (ESP32-S3) developed using PlatformIO. The boot flow initializes Wi-Fi and peripheral modules, starts the embedded web server, and enters the main loop for event handling. The firmware ensures automatic Wi-Fi reconnection to maintain continuous service availability. Also integrated a Wi-Fi manager that provides user friendly network connection. Firmware based on an asynchronous web server.

The ESP32-S3 acts as the core processing unit. It acquires video frames from the OV3660 camera, polls the PN532 for RFID tags when present, reads the HC-SR04 to prevent hazardous barrier motion, and exchanges data and commands with the backend over Wi-Fi.

The firmware follows a deterministic initialization and event-driven loop. Firstly, the firmware initializes storage and connectivity: mounts SPIFFS for static assets, configures Wi-Fi in AP or STA mode, and ensures automatic reconnection to maintain continuous availability. Then the firmware initializes peripherals: configure the OV3660 camera (frame size, pixel format and stability parameters), set up the PN532 over I2C for RFID reads, initialize the HC-SR04 for obstacle detection, and prepare PWM control for the servo.

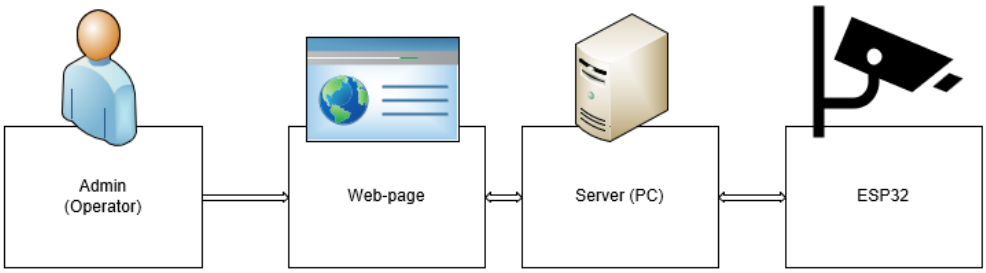


Fig. 1. System architecture, showing interactions between components.

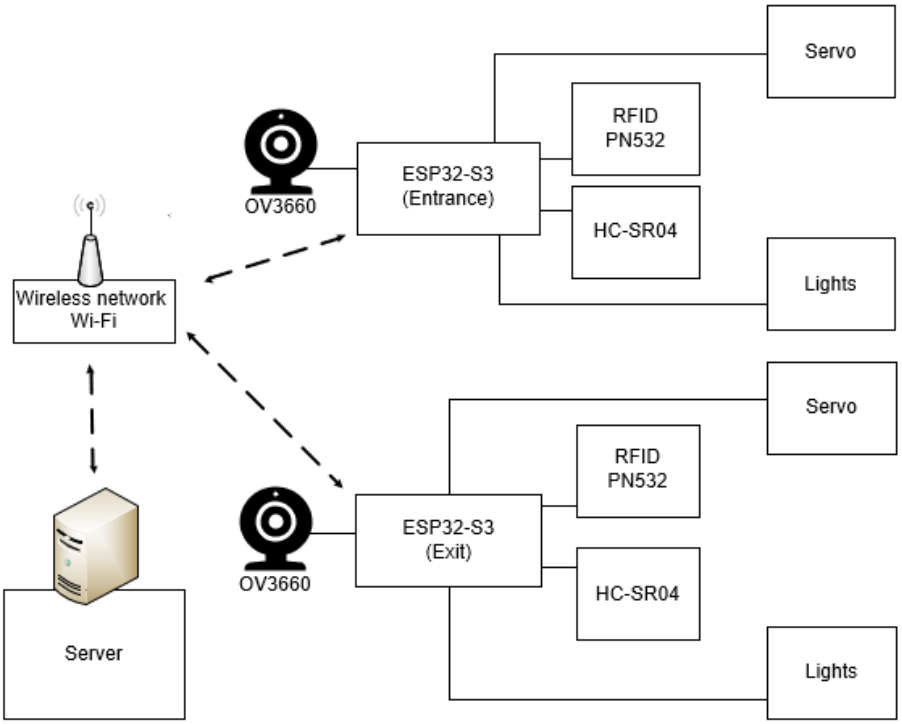


Fig. 2. System architecture, showing interactions between components.

Secondly, the firmware starts services: launch the embedded web server and REST endpoints used for device interaction and monitoring. Then enter the main loop that consists of capturing images and sending snapshots to the backend for recognition. Receiving access decisions (open/close) and actuating the barrier. Polling RFID events and forward them to backend routes for verification. Continuously log system activities and maintain health/status for diagnostics

The system operates as follows: when a vehicle approaches, the camera captures an image that the backend processes using a YOLOv8-based ALPR pipeline. The recognized plate is checked against an authorized (white) list in the database. If the plate is not authorized, the system supports manual authentication via the RFID module. A distance sensor enforces a safety interlock so the barrier only moves when it is safe to do so.

The backend is a critical component responsible for processing authentication requests and orchestrating access decisions. Implemented with Flask [13], it provides RESTful APIs for the microcontroller and the web interface, and uses YOLOv8 for ALPR, SQLite (via SQLAlchemy) for storage, and asynchronous threading for concurrent request handling **Fig. 3a**.

The backend receives image data from the ESP32 and pre-processes it (e.g., resizing, color conversions). Then runs the YOLOv8-based pipeline to detect vehicles and recognize license plates (detection + OCR in the region of interest) **Fig. 3b**. After matches the recognized plates against an authorized “whitelist” in the database. Logs all access attempts for security audits and exposes retrieval via API (e.g., /logs). Sends control commands (open/close) back to the ESP32 microcontroller over REST.

In addition to ALPR, the backend handles RFID-based events via dedicated endpoints, validates tags against the database, and logs outcomes for auditability. The use of RESTful APIs enables seamless communication with both the microcontroller and the web interface, while caching is employed to maintain responsiveness for real-time operation.

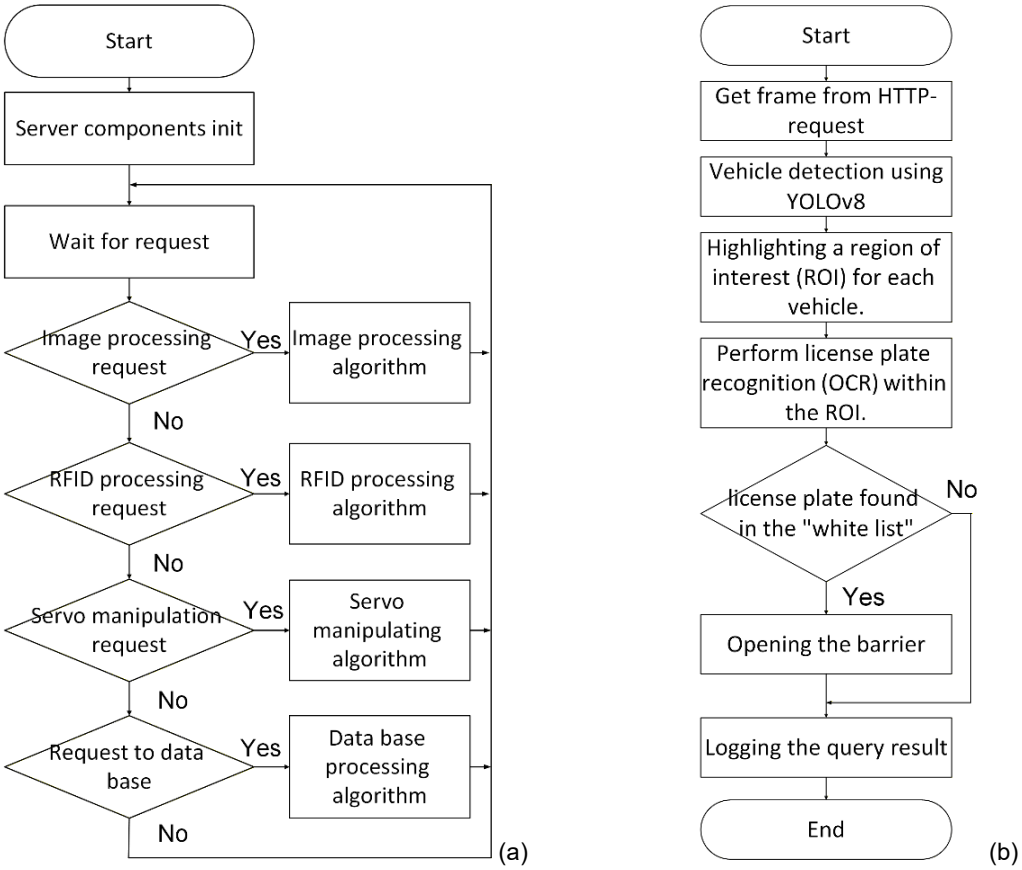


Fig. 3. The backend scheme (a) and the image processing algorithm (b).

The system logs every access attempt (ALPR and RFID), stores audit trails, and provides retrieval via API for operator review in the web interface. Configuration parameters (e.g., safe distance) are synchronized via API to keep device and server settings consistent in real time.

The web interface (React-based) enables real-time monitoring of camera feeds and logs, management of registered users/vehicles, system configuration (e.g., safe distance), and manual barrier control when required. The React-based web interface **Fig. 4** provides operators and administrators with a streamlined, real-time view of system activity and controls. It enables users to: view real-time access logs and monitor live camera feeds, with automatic reconnect attempts every five seconds to maintain continuity under network issues. Manage registered users, vehicles, and plate storage (e.g., “Cars and plates storage”), and review “Last cars” entries with timestamps, plate numbers, and associated info **Fig. 5**.

The interface implements authentication and session handling (e.g., isAuthenticated token and login redirect) and integrates with the Flask backend via RESTful APIs for log retrieval and device commands, supporting responsive real-time operation. Role-based access can be applied to ensure security personnel and administrators see only the features relevant to their responsibilities.

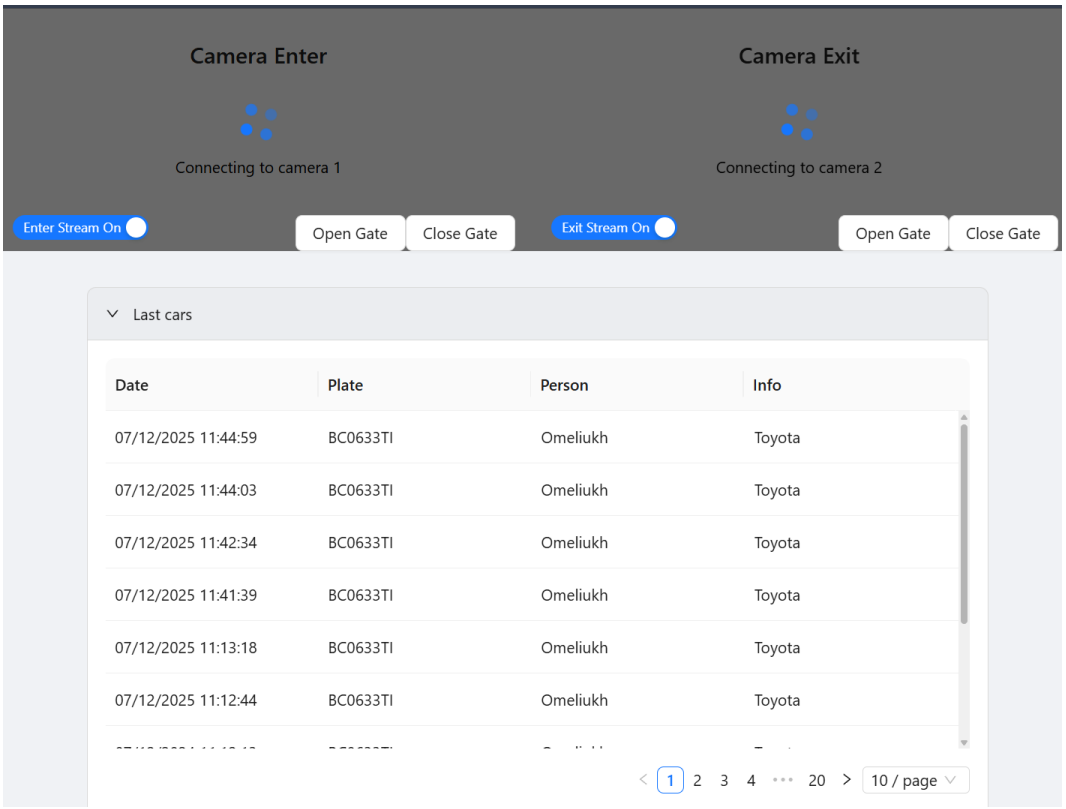


Fig. 4. Web interface screenshot displaying real-time vehicle monitoring and access control functions.

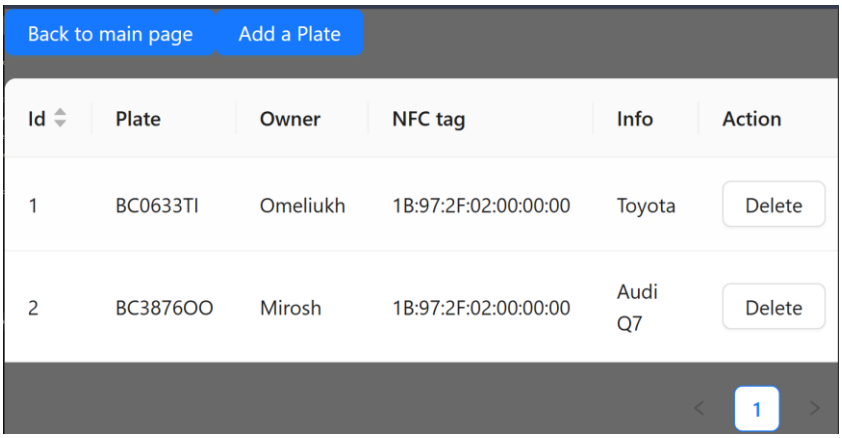


Fig. 5. Web interface "Cars and plates storage".

Adjust system settings and security parameters, including safe-distance thresholds for barrier operation (e.g., "Current Safe Distance: 20 cm"). Manually override barrier control when necessary for operational continuity **Fig. 6**.

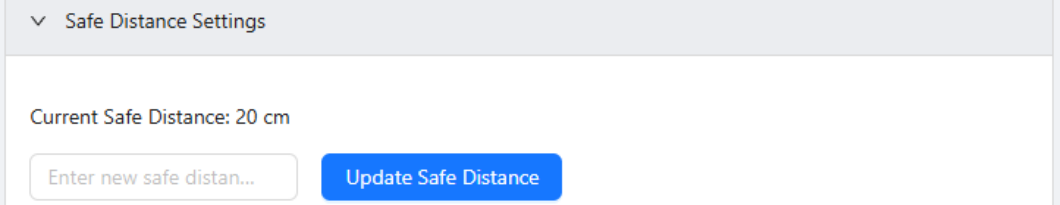


Fig. 6. Web interface. Safe-distance thresholds.

RESULTS AND DISCUSSION

To validate the performance and reliability of the proposed system, a series of controlled experiments was conducted. The evaluation focused on two primary metrics: the accuracy of the Automatic License Plate Recognition (ALPR) module and the overall system response time.

To determine the accuracy of the YOLOv8-based ALPR pipeline, a custom dataset consisting of 1,000 static images was curated. This dataset was designed to reflect real-world operating conditions and included images captured at various times of day (daylight, dusk, and night), different weather conditions (clear, overcast), diverse viewing angles, and with partially obscured or soiled license plates.

The evaluation was performed using a Python script that processed each image in the dataset. A "successful" recognition was defined as an instance where the system correctly identified and transcribed all alphanumeric characters on the license plate in the correct order.

Out of the 1,000 test images, the system successfully recognized 750 plates, resulting in an overall accuracy of 75%. An analysis of the 250 failed cases revealed that the majority of errors occurred under specific challenging conditions: Low Light/Night: 42% of failures were due to insufficient illumination or heavy headlight glare. Obscured Plates: 30% of failures involved plates partially covered by dirt, snow, or tow hitches. Extreme Angles: 20% of failures occurred when the vehicle approached the camera at an angle greater than 40 degrees. Other: 8% of failures were due to motion blur or non-standard fonts on the plates.

System response time was defined as the end-to-end latency, encompassing image acquisition by the OV3660, Wi-Fi transmission – using 2.4 GHz 802.11n, backend processing – ALPR inference and database lookup, and the return transmission of the control command to the ESP32-S3. This metric was recorded under stable network conditions with a signal strength (RSSI) better than –65 dBm. System response time was recorded for all 1,000 test cases in the accuracy assessment. The resulting data was analyzed to determine the central tendency and variance. The arithmetic mean response time for the system was 820 ms. The standard error ±45 ms, indicating consistent performance with low latency across the test batch. This sub-second response time confirms the system's suitability for real-time operation without causing significant delays for users.

In addition to the primary metrics, functional tests were performed on auxiliary components. The PN532 module's reading range was confirmed to be reliable up to 5 cm at various card presentation angles (0, 45, and 90 degrees). The HC-SR04 ultrasonic sensor was validated by repeatedly placing obstacles at the barrier's path, confirming that it correctly halted barrier motion in 100% of test cases. All performance data was logged by the Flask backend application and analyzed using custom Python scripts utilizing the Pandas and NumPy libraries for statistical computation.

A critical aspect of system optimization was identifying the optimal camera resolution. This parameter presents a fundamental trade-off: higher resolutions (e.g., UXGA, 1600x1200) provide superior image detail for the ALPR model but drastically reduce the real-time processing speed, measured in average frames per second (FPS). Conversely, lower resolutions (e.g., QVGA, 320x240) are processed very quickly but often fail to capture enough detail for accurate recognition, significantly reducing the system's 75% accuracy.

A systematic test was conducted to measure the average FPS the system could achieve when processing streams at different standard resolutions. The results of this analysis are illustrated in **Fig. 7**. As the graph clearly shows, there is a sharp, non-linear drop in performance as resolution increases.

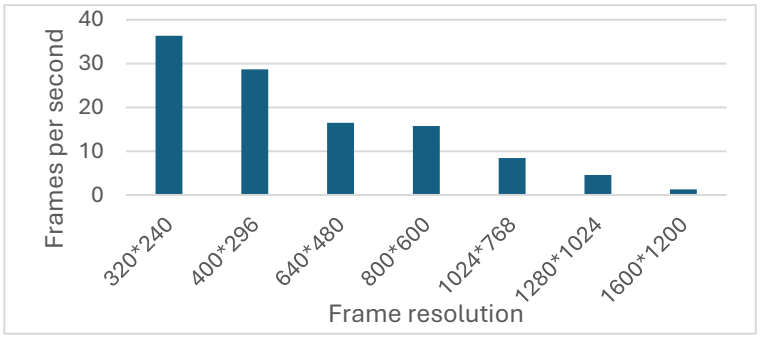


Fig. 7. Impact of image resolution on the system's average processing speed (FPS).

Based on this data, the 800x600 (SVGA) resolution was identified as the optimal trade-off. It provides a processing speed that is fast enough for responsive, real-time operation while preserving the image quality necessary for the ALPR model to function effectively.

CONCLUSION

This study successfully demonstrated the design, implementation, and practical validation of a cost-effective, automated access control system based on IoT principles. The key achievement of the work is the successful integration of a dual-modal authentication architecture, combining Automatic License Plate Recognition (ALPR) with a robust Radio-Frequency Identification (RFID) fallback, managed by a modern client-server backend.

The most significant results stem from the system's performance evaluation. The ALPR module, powered by a YOLOv8 model, achieved an accuracy of 75% on a challenging 1,000-image dataset. This quantitative result provides a realistic baseline for the system's effectiveness. Furthermore, the system demonstrated excellent real-time capabilities, with an average response time of 820 ms (±45 ms standard error), confirming its suitability for high-traffic environments where delays are unacceptable. The successful integration of an ultrasonic safety sensor, which reliably prevented barrier motion in 100% of test cases, underscores the system's readiness for practical deployment.

The practical significance of this research lies in its complete, end-to-end system design. By combining edge computing on the ESP32-S3 microcontroller with a flexible backend (Python Flask) and a real-time monitoring web interface (React), this work provides a scalable and affordable blueprint that significantly enhances security and efficiency compared to traditional manual access methods.

The analysis of ALPR failures (which led to 75% accuracy) clearly illuminates the path for future improvements. Failures were predominantly caused by adverse lighting conditions (headlight glare, low light) and significant plate obstructions. Therefore, future prospects for this research should focus on enhancing the robustness of the computer vision model. This can be achieved by re-training the model on a more diverse dataset featuring augmented night-time and poor-weather conditions, as well as exploring image pre-processing algorithms (e.g., local contrast enhancement) to improve image quality before recognition.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by the Ministry of Education and Science of Ukraine (Project No. 0125U001883).

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [N.O., H.K., T.T]; methodology, [N.O.]; investigation, [N.O.]; writing – original draft preparation, [N.O., H.K.]; writing – review and editing, [N.O., H.K.]; visualization, [N.O.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Bamashmos, S., Chilamkurti, N., & Shahraki, A. S. (2024). Two-Layered Multi-Factor Authentication Using Decentralized Blockchain in an IoT Environment. Sensors, 24(11), 3575. https://doi.org/10.3390/s24113575

[2] Babii, A., & Samila, A. (2023). Dual Authentication Technique for RFID Access Control Systems with Increased Level of Protection. Security of Infocommunication Systems and Internet of Things, 1(1), Article 01011. https://doi.org/10.31861/sisiot2023.1.01011

[3] Riad, K. (2025). Robust and Leakage-Resilient Access Control for IoT Outsourcing with Attribute-Based Encryption. Sensors, 25(3), 625. https://doi.org/10.3390/s25030625

[4] Albugmi, A. (2025). Hybrid smart IoT detection and prevention framework for smart cities using blockchain technology. International Journal of Advanced and Applied Sciences, 12(4), 107-115. https://doi.org/10.21833/ijaas.2025.04.013

[5] Wang, X., Wang, M., Guo, H., Li, J., Wang, X., & Zhang, Y. (2025). License plate recognition system for complex scenarios based on improved YOLOv5s and LPRNet. Scientific Reports, 15, Article number 34741. https://doi.org/10.1038/s41598-025-18311-4

[6] Fadlianda, D., Fikry, M., & Nunsina. (2024). Innovative IoT-Based Automatic Gate System with RFID and Electro-Magnetic Lock for Secure Access. Proceedings of Malikussaleh International Conference on Multidisciplinary Studies (MICoMS), 4, Article 884. https://doi.org/10.29103/micoms.v4i.884

[7] Pradhan, G., Prusty, M. R., Negi, V. S., Chinara, S., et al. (2025). Advanced IoT-integrated parking systems with automated license plate recognition and payment management. Scientific Reports, 15, Article number 2388. https://doi.org/10.1038/s41598-025-86441-w

[8] Kokila, M., & Srinivasa Reddy, K. (2024). Authentication, access control and scalability models in Internet of Things security – A review. Cyber Security and Applications, 10, 123-146. https://doi.org/10.1016/j.csa.2024.100057

[9] Kalamaras, S. D., Tsitsimpikou, M.-A., Tzenos, C. A., Lithourgidis, A. A., Pitsikoglou, D. S., & Kotsopoulos, T. A. (2025). A Low-Cost IoT System Based on the ESP32 Microcontroller for Efficient Monitoring of a Pilot Anaerobic Biogas Reactor. Applied Sciences, 15(1), 34. https://doi.org/10.3390/app15010034

[10] Chang, Y.-H., Wu, F.-C., & Lin, H.-W. (2025). Design and Implementation of ESP32-Based Edge Computing for Object Detection. Sensors, 25(6), 1656. https://doi.org/10.3390/s25061656

[11] Babii, A., & Samila, A. (2023). Dual Authentication Technique for RFID Access Control Systems with Increased Level of Protection. SISIOT, 1(1), 01011. https://doi.org/10.31861/sisiot2023.1.01011

[12] Syahputra, D., Adriansyah, A., & Wibowo, A. T. (2023). Performance sensor analysis of HC-SR04 proximity sensor on distance measuring device with fuzzy logic method. JEEMECS: Journal of Electrical Engineering, Mechatronic and Computer Science, 7(1), 1-7. https://doi.org/10.26905/jeemecs.v7i1.10096

[13] Sabbatini, M. (2024). Hardening IoT Devices: An Analysis of the ESP32-S3 Microcontroller. (Master's thesis). University of Zurich. https://doi.org/10.1109/ACCESS.2021.3092938

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ НА БАЗІ ІНТЕРНЕТУ РЕЧЕЙ З ІНТЕГРАЦІЄЮ ТЕХНОЛОГІЙ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ТА РАДІОЧАСТОТНОЇ ІДЕНТИФІКАЦІЇ

Назар Омелюх*, *Галина Клим**, *Тарас Ткачук
Національний університет «Львівська політехніка»,
вул. Бандери 12, 79013 м. Львів, Україна

АНОТАЦІЯ

Обґрунтування. Забезпечення безпеки приватних територій стає дедалі актуальнішим завданням у сучасному світі. Традиційні методи контролю, що покладаються на фізичну охорону або механічні бар'єри з використанням ключів, є малоефективними та вразливими до ризиків, таких як крадіжка або дублювання перепусток. Крім того, застарілі системи не забезпечують належного аудиту подій. Це формує нагальну потребу у сучасних автоматизованих IoT-системах, які гарантують надійне управління доступом та моніторинг у реальному часі.

Матеріали та методи. Ядром системи є недорогий мікроконтролер з модулем камери. Зчитувач радіочастотної ідентифікації (RFID) сканує картки доступу. Ультразвуковий сенсор відстані виявляє перешкоди для безпеки, а сервопривід керує фізичним шлагбаумом. Серверне програмне забезпечення (бек-енд) було розроблено на Python, а веб-панель керування – на JavaScript (React).

Система поєднує два методи ідентифікації. По-перше, камера фіксує зображення транспортного засобу, надсилаючи його на сервер, де модель ШІ зчитує номерний знак. Сервер перевіряє номер у списку дозволених. Якщо номер не розпізнано, водій сканує RFID-картку як вторинний метод. Сенсор відстані постійно контролює зону шлагбаума, щоб запобігти закриттю на перешкоді. Веб-інтерфейс дозволяє оператору спостерігати за камерою, переглядати журнали та вручну управляти шлагбаумом.

Результати. Розроблена система була успішно протестована. Модель ШІ досягла 75% точності в ідентифікації номерних знаків. Система виявилася швидкою, із середнім часом відгуку від зйомки до прийняття рішення менше однієї секунди. Датчик безпеки був перевірений, надійно виявляючи перешкоди та запобігаючи руху шлагбаума, що забезпечує безпечну роботу. Були представленні результати порівняння якості відео та часом відгуку системи. Оптимальний баланс між якістю відео та часом відгуку було знайдено при роздільній здатності 800x600.

Висновки. Було успішно розроблено, побудовано та протестовано надійну, економічно ефективну автоматизовану систему контролю доступу. Поєднання зчитування номерних знаків на основі ШІ з резервною системою RFID забезпечує надійне та гнучке рішення. Ця система добре підходить для покращення безпеки та ефективності в реальних умовах, таких як житлові, офісні та промислові зони.

Ключові слова: автоматизований контроль доступу, ESP32-S3, ALPR, RFID, безпека.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
28 November, 2025	15 December, 2026	16 December, 2025	25 December, 2025

Elecronics and information technologies, 2025, 32, 141–150
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 621.382, 537.312

A QUANTITATIVE ANALYSIS OF WEBASSEMBLY INTEGRATION: ARCHITECTURAL PATTERNS, TOOLING, AND PERFORMANCE EVALUATION

Oleksandr Stepanov , ***Halyna Klym**** ,
Lviv Polytechnic National University
12 Bandera St., Lviv 79000, Ukraine

Stepanov O. et al. (2025). A Quantitative Analysis of WebAssembly Integration: Architectural Patterns, Tooling, and Performance Evaluation. *Electronics and Information Technologies*, 32, 141–150. https://doi.org/10.30970/eli.32.10

ABSTRACT

Background. WebAssembly (Wasm) is a fundamental component for high-performance web applications, valued for strategic integration, not simple JavaScript replacement. Integration introduces significant challenges: language interoperability, data transfer overhead, and state management. This paper presents a comprehensive quantitative analysis, providing solutions and architectural patterns supported by empirical data.

Materials and Methods. The study comprised two parts. Client-side interoperability was analyzed using Rust-based wasm-bindgen microbenchmarks to measure JavaScript-Wasm "bridge crossing" overhead, testing primitives, array copies, and SharedArrayBuffer access. Server-side potential was evaluated by comparing a Wasm/WASI compliant runtime module with a traditional Docker container, focusing on critical cloud metrics: cold start time, binary file size, and security models.

Results and Discussion. Interoperability costs vary significantly. Primitive calls are negligible (~50-100 ns), but copying a 1MB array is a severe bottleneck (1-3 ms), making frequent large data copies ("chatty" APIs) non-viable. SharedArrayBuffer overhead is minimal (~15 ns). Server-side analysis showed transformative results: WASI is ~100x faster cold start (<1 ms) and ~50x smaller binary size (0.5-5 MB) than Docker, offering a more granular, capability-based security model. Benchmarks confirm Rust+Wasm achieves up to 8.7x performance gains. We discuss "Wasm as a Pure Function" vs. "Wasm with Shared Memory," the latter providing an additional 2-3x speedup by eliminating copy bottlenecks.

Conclusion. Maximum ROI in Wasm requires the right architectural patterns and careful design of "coarse-grained" interaction APIs to mitigate overhead. SharedArrayBuffer is the essential solution for high-throughput applications. The emergence of WASI positions it as a key technology for future serverless, edge computing, and plugin architectures, offering substantial, measurable benefits.

Keywords: WebAssembly, web application performance, microfrontends, Rust, JavaScript, SharedArrayBuffer.

INTRODUCTION

Modern web applications require ever-increasing amounts of processing power, often beyond the capabilities of traditional JavaScript. While JavaScript is a powerful JIT-compiled language, its interpreted and dynamically typed nature can create bottlenecks in CPU-intensive tasks, such as image processing, physics simulations, or real-time big data analysis.WebAssembly (Wasm) was created as an answer to this challenge. It is a binary instruction format designed to execute efficiently in a web environment, allowing for near-

© 2025 Oleksandr Stepanov & Halyna Klym. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

native speeds. Wasm has finally moved from an experimental technology to a fundamental tool for building a new generation of web applications.The key idea, however, is that the true value of Wasm is not revealed through a complete replacement of JavaScript, but through deep and thoughtful integration. This article goes beyond theoretical promises to demonstrate the measurable impact of Wasm on the performance and architecture of modern web systems. We provide a comprehensive analysis of challenges related to language interaction, DOM access, and state management, and propose concrete solutions supported by quantitative data and architectural patterns [1].

The increasing complexity of modern web applications places demands on computing power that often exceed the capabilities of traditional JavaScript. While WebAssembly (Wasm) is positioned as a solution to this problem, offering near-native performance, its effective implementation remains a non-trivial task. Despite the theoretical advantages, there is a lack of systematic analysis of the practical aspects of deep Wasm integration. Developers face bottlenecks in interoperability, where the overhead of calls between JavaScript and Wasm can negate any performance gains. In addition, architectural ambiguity and the lack of clear, proven patterns for integrating Wasm modules into modern architectures, such as microfrontends, lead to development complexity. Many decisions to use Wasm are made based on general promises rather than specific quantitative metrics, making it difficult to assess the return on investment. Thus, the problem lies in the lack of a comprehensive study that would quantify the benefits of Wasm, systematize the architectural patterns of its integration, and provide developers with clear recommendations for minimizing overhead.

The main goal of this research is to conduct a comprehensive analysis of the deep integration of WebAssembly into modern web applications to determine the most effective strategies and architectural patterns. To achieve this goal, the research will quantify the performance gains from using Rust+Wasm modules compared to optimized JavaScript, and also analyze and systematize architectural patterns for Wasm integration into microfrontends. The work will investigate the overhead of interaction between JavaScript and Wasm for different data types, including shared memory, and assess the potential of WebAssembly System Interface (WASI) as a server technology compared to Docker containers. Based on the data obtained, practical recommendations will be formulated for developers on choosing tools and designing APIs to achieve maximum performance [2-3].

ANALYSIS OF JS-WASM INTEROPERABILITY OVERHEAD

To achieve the stated research goal, two key experimental studies were conducted. The first focused on quantitatively measuring the overhead associated with the interaction between JavaScript (JS) and Wasm. The second focused on evaluating the potential of the WebAssembly System Interface (WASI) as a server-side technology in comparison to the dominant containerization technology, Docker. One of the most critical, yet often underestimated, aspects of deep WebAssembly integration is the cost of "bridge crossing", the overhead incurred with every function call between the JS and Wasm environments. The efficiency of the integration, and thus the overall performance gain, is directly dependent on minimizing these costs. This study aimed to quantitatively measure and analyze this overhead for various data types and transfer methods. To measure this, a set of microbenchmarks was developed to simulate four common interaction scenarios [4]. To ensure statistical reliability and mitigate the impact of system jitter and JIT (Just-In-Time) compilation variability, each microbenchmark scenario was subjected to a rigorous testing protocol. The tests were executed with N = 10,000 iterations. To account for the "warm-up" period required by the JavaScript engine to optimize hot paths, the initial 1,000 iterations were discarded. The results presented in **Table 1** represent the arithmetic mean of the remaining stable iterations. Outliers deviating more than three standard deviations from the mean were excluded to prevent temporary system background processes from skewing

the data. The testing was conducted using a Wasm module compiled from Rust and the wasm-bindgen toolchain. This choice was deliberate, as wasm-bindgen is the standard for the Rust ecosystem, automating the generation of interoperability code and managing the Wasm module's linear memory to ensure correct data type conversion [5, 6]. The average time (in nanoseconds) was measured for calls under the following scenarios: transferring primitive types (integer arguments), transferring string data (a short string), copying large data volumes (a 1MB array), and accessing shared memory (SharedArrayBuffer). The measurement results, presented in **Table 1**, demonstrate significant variability in overhead depending on the data type.

Table 1. **Overhead of JS → Wasm Calls (nanoseconds)**

Call	TypeOverhead (ns)	Description
JS → Wasm (integer arguments)	50 - 100 ns	Very low overhead, comparable to a native JS function call.
JS → Wasm (short string)	600 - 2,500 ns	Higher overhead due to encoding (JS: UTF-16, Wasm: UTF-8) and memory allocation.
JS → Wasm (copying 1MB array)	1,000,000 - 3,000,000 ns (1-3 ms)	Cost is dominated by memory copy time; can become a bottleneck.
JS ↔ Wasm (shared memory access)	~15 ns	Extremely low overhead, as no copying is involved. Direct memory access.

An analysis of the obtained data underscores the critical importance of careful API design for the interaction between JavaScript and WebAssembly. As a practical calculation shows, the impact of this overhead is directly dependent on the frequency and volume of calls. If an application makes 1,000 calls per second passing short strings, the total overhead will be up to 2.5 ms per second (2,500 ns * 1,000), which is negligible and does not affect overall performance. This scenario is suitable for "chatty" APIs where small pieces of data are frequently exchanged. However, the situation changes drastically when working with large data volumes. If each of those 1,000 calls copies a 1 MB array, the total overhead increases to 3 seconds (3 ms * 1,000). This creates a paradoxical situation where the data transfer overhead completely negates any benefits from fast computation in Wasm. This proves that an architecture relying on frequent copying of large buffers is non-viable [7-9]. The results clearly indicate that for tasks requiring the processing of tens of megabytes of data per frame (e.g., real-time video editing, WebGL rendering, or complex scientific analysis), the only effective solution is the use of SharedArrayBuffer [10]. This approach, although more complex to implement due to the need for manual memory management and the risk of race conditions, eliminates the bottleneck associated with copying and, as tests show, can provide an additional 2-3x acceleration compared to the copying method.

EVALUATION OD WASI FOR SERVER-SIDE COMPUTING

In addition to accelerating web applications, this research also examined the potential of WebAssembly outside the browser, specifically through the WebAssembly System Interface (WASI) standard. WASI transforms Wasm into a universal, secure, and portable binary format, allowing Wasm modules to run on servers, in cloud environments, and on edge devices, which is a strategically important direction for the technology's development. A comparative analysis was conducted on key performance and security metrics for two technologies: a Wasm module running in a WASI-compliant runtime, and a traditional Linux

container run via Docker, which is currently the industry standard for deploying server-side applications [11]. The comparison was based on four key metrics critical for modern cloud and serverless architectures: cold start time, binary size, security, and portability.

To ensure the reliability and reproducibility of the comparative data presented, a strict measurement protocol was adopted. The cold start time and binary size benchmarks were repeated N = 500 times for both the Wasm/WASI environment (using the Wasm time runtime) and the Docker container environment (using an Alpine Linux base image). For the cold start metric, the system was reset between iterations to eliminate caching effects, ensuring 'true' cold start conditions. The values presented in **Table 2** represent the arithmetic mean of these 500 iterations, with the highest and lowest 5% of results excluded to filter out transient system latency spikes.

The quantitative comparison of these metrics revealed fundamental advantages of Wasm/WASI for server-side computing, as shown in **Table 2**.

Table 2. **Comparative Analysis: Wasm/WASI vs. Docker Containers**

Metric	Wasm with WASI	Docker Container	Wasm Advantage
Cold Start Time	< 1 ms	100 - 500+ ms	~100x Faster
Binary Size	0.5 - 5 MB	50 - 500+ MB	~50x Smaller
Security	Sandboxed by default, granular resource access.	OS-level virtualization, larger attack surface.	Significantly Higher
Portability	Universal (any runtime)	OS/Architecture (x86, ARM)	Absolute

The obtained metrics demonstrate that Wasm/WASI is an ideal technology for a new generation of cloud architectures, particularly for serverless functions, service mesh plugins, and IoT devices. The advantage in cold start time (approximately 100-fold) is transformative. While a Docker container requires hundreds of milliseconds to initialize a virtualized OS environment and start a process, a Wasm module starts in under a millisecond, as it is simply a lightweight process within an existing runtime [12, 13]. This allows for the implementation of true "pay-per-request" models without the need to keep resources "warm." Combined with the minimal binary size (approximately 50-fold smaller), this provides unprecedented density and efficiency of computation. Where one Docker container could run on a server, hundreds of isolated Wasm modules could potentially run simultaneously. A key difference is also the security model. Docker relies on OS-level virtualization, which, while robust, leaves a large attack surface. WASI, in contrast, uses a capability-based security model (**Fig.1**).

This means a Wasm module, by default, has access to nothing — not the file system, not the network, not even the system clock. It must be granted permissions granularly (e.g., access to a specific folder or socket), which drastically reduces potential damage in case of a compromise and significantly increases the overall security posture.

DISCUSSION AND ARCHITECTURAL IMPLICATIONS

The analysis of the empirical data gathered in this study allows for a deep interpretation of the practical implications and strategic advantages of WebAssembly integration. This discussion extends beyond the mere statement of performance gains to cover architectural dilemmas, tooling trade-offs, and the transformative potential of the technology beyond the browser. The key finding from the quantitative analysis (presented in **Table 1**) is that the performance gain from implementing WebAssembly is non-linear and profoundly dependent on the nature of the task. As the calculations show, the most

significant effect (ranging from 6.4x to 8.7x) is observed in tasks requiring complex mathematical computations and intensive manipulation of large data volumes in memory. The reason for this significant gap is fundamental: Wasm is a binary instruction format designed for efficient execution using static typing and a linear memory model, which avoids the unpredictable pauses associated with JavaScript's Garbage Collector (GC). In contrast, even a highly optimized Just-In-Time (JIT) compiler for JavaScript must contend with a dynamically typed language, which introduces significant overhead for type-checking and on-the-fly optimizations [14]. The physics simulation is the most telling example: the Rust+Wasm module processes approximately 5.2 million operations per second, whereas the JavaScript version reaches only 600,000. This eightfold acceleration is not merely an incremental improvement; it is a qualitative leap that fundamentally changes the web development paradigm. It opens the door for an entire class of applications previously considered impossible to implement in a browser and were the exclusive domain of desktop programs: full-fledged computer-aided design (CAD) systems, real-time 3D game engines, and tools for interactive scientific modeling and data analysis. For the end-user, this translates to immediate interactivity and a lag-free experience in web interfaces that was previously unattainable.

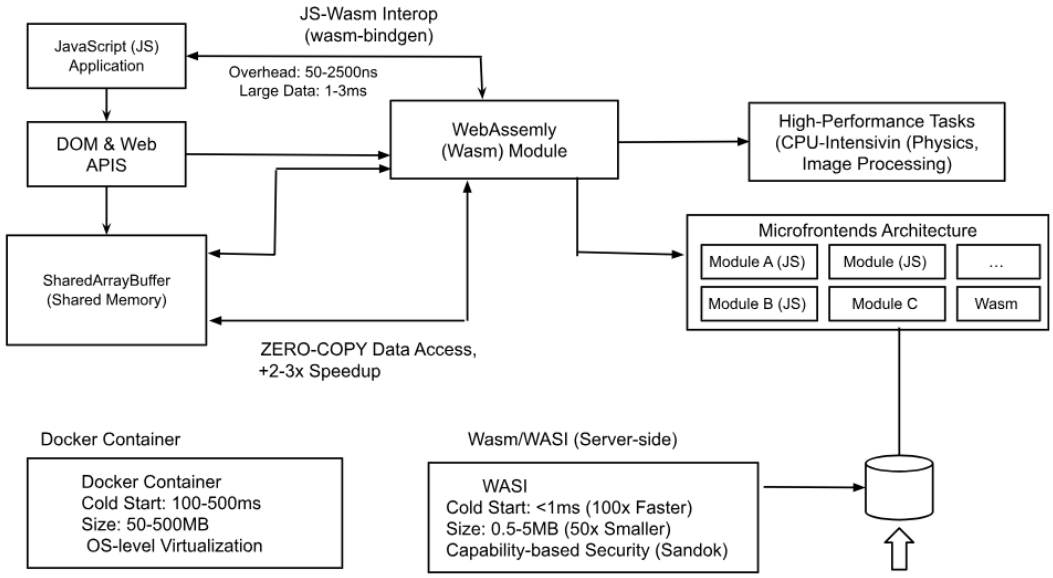


Fig.1. Client-Side WASM Module Compilation and Execution Architecture.

The findings also confirm that a micro-frontend architecture is an exceptionally effective model for implementing WebAssembly [15, 16]. It allows for the encapsulation of compute-intensive logic into isolated Wasm modules, transforming them into high-performance "black boxes." This approach not only accelerates specific functionality but also improves overall application stability. The choice of the correct integration pattern, however, is key to achieving a balance between performance and development complexity. The "Wasm as a 'Pure Function'" pattern is the simplest to implement and test, ideal for algorithms, mathematical calculations, and validation logic. However, its simplicity is deceptive; the pattern is limited by data I/O, as every interaction requires serialization and deserialization of the data being passed, creating overhead that can become a bottleneck if calls are frequent and data volumes are significant. In contrast, using Wasm with shared memory (SharedArrayBuffer) is the most performant but also the most architecturally

complex approach. It completely eliminates the data-copying bottleneck by allowing JavaScript and Wasm to operate on the same block of memory. For tasks requiring the processing of tens of megabytes of data per frame, such as real-time video effects, this approach can provide an additional 2-3x acceleration on top of the baseline Wasm gain [17]. This power comes at a high cost: developers are forced to manage memory manually and implement complex synchronization mechanisms (e.g., using Atomics) to avoid race conditions. Finally, the "Wasm as a Full Component" pattern is a compromise, providing complete encapsulation of logic and state. It is optimal for complex, self-contained widgets (like an embedded 3D editor), but its disadvantage is the significant amount of "glue" code required in JS to bridge events, data, and APIs, which increases initial development complexity.

The data on the cost of calls between JS and Wasm (presented in **Table 2**) is perhaps the most critical finding for practicing architects. It underscores the vital importance of careful API design. A naive approach, where a Wasm module is treated as a simple replacement for a JavaScript function, can lead to a catastrophic performance drop. The example of copying a 1MB array, where the overhead for 1,000 calls can reach 3 seconds, vividly illustrates this trap. This completely negates the Wasm advantage, even if the computation itself takes only milliseconds. This implies that Wasm APIs must be "coarse-grained"- a single call should perform as much work as possible—rather than "chatty." This is where tooling like wasm-bindgen, the standard for the Rust ecosystem, comes into play. It automates the generation of interop code, managing Wasm's linear memory and handling complex tasks like converting JS (UTF-16) strings to Wasm (UTF-8). However, this convenience has a price, as shown by the higher cost of string transfers. Developers face a clear trade-off: use the convenient abstractions of wasm-bindgen and pay a performance penalty, or write custom glue code with manual memory management to achieve maximum speed.

Finally, the research confirms that Wasm's potential extends far beyond the browser. The metrics from **Table 2** position Wasm/WASI as an ideal technology for serverless functions, service mesh plugins, and IoT devices [18]. The reduced cold start time (under 1 ms) and minimal binary size (up to 5 MB) enable an unprecedented density and efficiency of computation in cloud environments. For cloud providers, this means the ability to run orders of magnitude more functions on the same hardware, leading to significant cost reductions. However, the most revolutionary advantage is the capability-based security model. Unlike Docker containers, which virtualize an entire OS and have a large attack surface, a Wasm module, by default, runs in a complete sandbox with no access to the file system, network, or system resources. The host environment must explicitly grant the module every single capability (e.g., "allow reading /config.toml"), providing a far superior level of granular security[19]. This makes WASI the ideal candidate for executing third-party code (e.g., in plugins) in a safe, isolated environment.

CONCLUSION

The deep integration of WebAssembly into modern web applications represents a paradigm shift, moving beyond experimental novelty to become a pragmatic and strategically advantageous solution for specific, high-impact computational challenges. The ecosystem surrounding Wasm, including mature tooling like wasm-bindgen for seamless Rust integration and evolving standards such as the WebAssembly System Interface (WASI), has reached a level of sophistication that makes robust integration not only feasible but increasingly compelling. This research quantitatively and irrefutably demonstrates that for CPU-intensive operations, spanning image processing, physics simulations, large-scale data manipulation, and complex parsing tasks, WebAssembly delivers performance gains often exceeding an order of magnitude compared to even highly optimized JavaScript implementations. These findings firmly shift the development

focus from the foundational question of possibility ("Can we do it?") towards the strategic consideration of application ("Where should we apply it for maximum return on investment?"). The decision to leverage Wasm should be driven by a clear understanding of the computational bottlenecks within an application and the specific performance goals.

However, realizing the full potential of this technology necessitates more than simply compiling existing code to Wasm. The choice of architectural integration pattern, whether treating Wasm as a pure function, leveraging shared memory via SharedArrayBuffer, or encapsulating logic within a full Wasm component, is paramount. Each pattern presents distinct trade-offs between raw performance, development complexity, and ease of testing, requiring careful consideration based on the specific use case. Based on our quantitative analysis, we recommend the following heuristic for real-world projects: developers should prioritize the 'Pure Function' pattern for isolated, stateless algorithmic tasks (such as cryptography or parsing) to maintain architectural simplicity; conversely, the 'Shared Memory' approach should be adopted strictly for high-throughput, real-time scenarios (like video processing) where data copying overhead becomes prohibitive, while the 'Full Component' model is best reserved for encapsulating complex, self-contained UI widgets or subsystems.

Furthermore, this study highlights the critical importance of meticulously designing the interface between JavaScript and WebAssembly. The non-trivial overhead associated with crossing the JS-Wasm boundary, particularly when transferring large or complex data structures like strings and arrays via copying, can significantly diminish or even negate the computational speed advantages if not managed effectively.

Architectures must favor coarse-grained APIs over chatty interactions, and for data-intensive applications, embracing the complexities of SharedArrayBuffer often becomes a necessity rather than an option. Simultaneously, the emergence of WASI underscores Wasm's transformative potential beyond the browser, offering substantial benefits in cold start times, binary size, and security posture compared to traditional containerization methods like Docker, positioning it as a key technology for future serverless, edge computing, and plugin architectures. Ultimately, unlocking the full, transformative power of WebAssembly hinges upon a holistic approach that combines quantitative performance analysis with informed architectural decisions and careful attention to the nuances of cross-language interaction.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The author(s) received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that there are no financial or other potential conflicts of interest regarding this work.

AUTHOR CONTRIBUTIONS

Conceptualization, [H.K., O.S.]; methodology, [H.K., O.S.]; investigation, [H.K., O.S.]; writing – original draft preparation, [O.S.]; writing – review and editing, [H.K., O.S.]; visualization, [O.S.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Schmidt, A., & Kovacs, L. (2024). High-performance AI in composable web architectures: A WebAssembly and micro-frontend approach. *Proceedings of the 2024 ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC)*, Pisa, Italy, 212–223. https://doi.org/10.1145/3658235.3658251

[2] Stepanov, O., & Klym, H. (2024). Features of the implementation of micro-interfaces in information systems. *Advances in Cyber-Physical Systems*, 9(1), 54–60. https://doi.org/10.23939/acps2024.01.054
[3] Stepanov, O., & Klym, H. (2024). Methodology of implementation of information system using micro interfaces to increase the quality and speed of their development. Computer Systems and Networks, 6(2), 222–231. https://doi.org/10.23939/csn2024.02.222
[4] Dubois, M., & Moreau, C. (2024). Dynamic loading and execution of AI models in micro-frontends using the WASM component model. Proceedings of the 2024 ACM SIGPLAN International Conference on Compiler Construction (CC), Edinburgh, UK, 78–89. https://doi.org/10.1145/3642939.3642947
[5] Brandt, L., & Sørensen, K. (2024). Seamless user experience: Combining lazy-loading of micro-frontends with streaming instantiation of WebAssembly AI modules. IEEE Software, 41(2), 30–37. https://doi.org/10.1109/MS.2023.3323210
[6] Costa, G., & Ferreira, M. (2024). WebGPU and WebAssembly: The next frontier for high-performance 3D and AI integration in composable web applications. Proceedings of the 29th International ACM Conference on 3D Web Technology (Web3D), San Sebastian, Spain, 1–10. https://doi.org/10.1145/3653481.3653488
[7] Szymański, M., & Nowak, A. (2024). Improving developer experience: A toolchain for debugging and profiling WebAssembly-based AI components in micro-frontend systems. Proceedings of the ACM/IEEE 4th International Workshop on Software Engineering for Web-Based Systems (SEW '24), Lisbon, Portugal, 67–74. https://doi.org/10.1145/3643750.3643758
[8] Moreau, F., & Bianchi, E. (2024). A hybrid execution model for web-based AI: Orchestrating client-side WASM and server-side GPU inference in micro-frontends. Proceedings of The Web Conference (WWW '24), Singapore, 1123–1134. https://doi.org/10.1145/3589334.3645657
[9] Stepanov, O., & Klym, H. (2024). Challenges, communication and future of micro frontends development and implementation. Proceedings of the 14th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 1–5. https://doi.org/10.1109/DESSERT65323.2024.11122150
[10] Shaik, S. (2025). Leveraging WebAssembly in micro frontend architectures: A technical deep dive. Journal of Computer Science and Technology Studies, 7(3), 860–865. https://doi.org/10.32996/jcsts.2025.7.3.95
[11] De Palma, G., Giallorenzo, S., Mauro, J., Trentin, M., & Zavattaro, G. (2024). FunLess: Functions-as-a-service for private edge cloud systems. Proceedings of the 2024 IEEE International Conference on Web Services (ICWS), Shenzhen, China, 41–51. https://doi.org/10.1109/ICWS62873.2024.00016
[12] Mathew, P. (2025). Front-end performance optimization for next-generation digital services. Journal of Computer Science and Technology Studies, 7(4), 993–1004. https://doi.org/10.32996/jcsts.2025.7.4.111
[13] Zhang, Y., Liu, M., Wang, H., Ma, Y., Huang, G., & Liu, X. (2024). Research on WebAssembly runtimes: A survey. ACM Transactions on Software Engineering and Methodology, 34(1), Article 29, 1–46. https://doi.org/10.1145/3639198
[14] da Silva, N. P. S., Rodrigues, E., & Conte, T. (2025). A catalog of micro frontends anti-patterns. IEEE Software, 42(1), 31–38. https://doi.org/10.1109/MS.2024.3468901
[15] Borello, D. (2024). Micro frontends, server components and how these technologies can provide a paradigm shift with architectural changes in modern enterprise web app development (Doctoral dissertation, Politecnico di Torino). https://doi.org/10.6092/polito/porto/1183181
[16] Lehman, T. J., & Ko, R. K. L. (2024). Wasm-bpf: A framework for WebAssembly-based BPF programs. Proceedings of the 2024 IEEE 21st International Conference

on Software Architecture (ICSA), Hyderabad, India, 13–24. https://doi.org/10.1109/ICSA59381.2024.00012
[17] Al-Azzoni, I. S., & Al-Husainy, M. A. F. (2024). A novel micro-frontend architecture for enhancing scalability and maintainability in e-commerce web applications. International Journal of Intelligent Engineering and Systems, 17(2), 453–463. https://doi.org/10.22266/ijies2024.0430.40
[18] Carrera, J. M., & Mazzeo, A. (2024). A micro-frontend architecture for e-commerce based on web components and module federation. Proceedings of the 2024 IEEE International Conference on E-Business Engineering (ICEBE), Florence, Italy, 248–253. https://doi.org/10.1109/ICEBE63920.2024.10660721
[19] Le, D. S., & Nguyen, P. H. H. (2024). Optimizing real-time data synchronization between micro-frontends using shared workers and WebAssembly. *International Journal of Web Information Systems, 20*(5), 714–735. https://doi.org/10.1108/IJWIS-04-2024-0045

КІЛЬКІСНИЙ АНАЛІЗ ІНТЕГРАЦІЇ ВЕБ-ЗБІРКИ: АРХІТЕКТУРНІ ШАБЛОНИ, ІНСТРУМЕНТИ ТА ОЦІНКА ПРОДУКТИВНОСТІ

Степанов Олександр **, *Галина Клим**
Національний університет «Львівська Політехніка»
вул. Бандери, 12, 79000 м. Львів, Україна

АНОТАЦІЯ

Вступ. WebAssembly (Wasm) – це фундаментальний компонент для високопродуктивних веб-застосунків, цінний для стратегічної інтеграції, а не простої заміни JavaScript. Інтеграція створює значні проблеми: мовну сумісність, накладні витрати на передачу даних та управління станом. У цій статті представлено комплексний кількісний аналіз, що пропонує рішення та архітектурні шаблони, підтверджені емпіричними даними.

Матеріали та методи. Дослідження складалося з двох частин. Взаємодія на стороні клієнта була проаналізована за допомогою мікробенчмарків wasm-bindgen на основі Rust для вимірювання накладних витрат на "перетин мосту" між JavaScript та Wasm, тестування примітивів, копій масивів та доступу до SharedArrayBuffer. Потенціал на стороні сервера був оцінений шляхом порівняння модуля середовища виконання, сумісного з Wasm/WASI, з традиційним контейнером Docker, зосереджуючись на критичних хмарних метриках: час холодного запуску, розмір бінарного файлу та моделі безпеки.

Результати та обговорення. Витрати на сумісність суттєво різняться. Виклики примітивів незначні (~50-100 нс), але копіювання масиву розміром 1 МБ є серйозним вузьким місцем (1-3 мс), що робить часті копії великих даних ("балакаючі" API) нежиттєздатними. Накладні витрати SharedArrayBuffer мінімальні (~15 нс). Аналіз на стороні сервера показав трансформаційні результати: WASI приблизно в 100 разів швидший за холодний старт (<1 мс) та приблизно в 50 разів менший за розміром двійкового файлу (0,5-5 МБ), ніж Docker, пропонуючи більш детальну модель безпеки, що базується на можливостях. Тести підтверджують, що Rust+Wasm досягає приросту продуктивності до 8,7 разів. Ми обговорюємо "Wasm як чисту функцію" проти "Wasm зі спільною пам'яттю", причому останній забезпечує додаткове прискорення в 2-3 рази, усуваючи вузькі місця копіювання.

Висновки. Максимальна рентабельність інвестицій (ROI) у Wasm вимагає правильних архітектурних шаблонів та ретельного проектування "грубозернистих" API взаємодії для зменшення накладних витрат. SharedArrayBuffer – це необхідне рішення

для високопродуктивних додатків. Поява WASI позиціонує його як ключову технологію для майбутніх безсерверних, периферійних обчислень та архітектур плагінів, пропонуючи суттєві, вимірювані переваги. Ключові слова: WebAssembly, продуктивність веб-додатків, мікрофронтенди, Rust, JavaScript, SharedArrayBuffer.

Ключові слова: WebAssembly, продуктивність веб-застосунків, мікрофронтенди, Rust, JavaScript, SharedArrayBuffer.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
20 November, 2025	17 December, 2025	17 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 151–170
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.4

DEFINITION AND FORMALIZATION OF THE SOFTWARE FUNCTIONAL STATE CONCEPT THROUGHOUT THE DEVELOPMENT LIFE CYCLE

Mariia Lyashkevych* , ***Vasyl Lyashkevych*** , ***Roman Shuvar***
Department of System Design
Faculty of Electronics and Computer Technologies
Ivan Franko National University of Lviv,
50 Drahomanova Str., 79005 Lviv, Ukraine

Lyashkevych, M.Y., Lyashkevych, V.Y., & Shuvar, R.Y. (2025). Definition and Formalization of the Software Functional State Concept Throughout the Development Life Cycle. *Electronics and Information Technologies*, 32, 151–170. https://doi.org/10.30970/eli.32.11

ABSTRACT

Background. Today, software is a critically important component of any information system. Its development requires significant resources and complex technical solutions, and the development of technologies is so rapid that not all concepts and definitions in the field of software are clearly formalized. This is especially true for the software functional state (SFS) throughout the software development life cycle (SDLC), as predicting all possible states is virtually impossible due to the dynamic nature of environments, changing requirements, component interactions, and the behavior of project participants. This creates a challenge for formalizing, analyzing, forecasting, monitoring, and managing these states.

Materials and Methods. The definition and formalization of SFSs encompass concepts from state theory in computer science, as well as quality models from international standards ISO/IEC 25010:2011 and the State Standard of Ukraine ISO/IEC 9126-1:2005. The defined concepts of SFS and SFS during SDLC are formalized mathematically, which allows building dynamic models of state evolution during SDLC based on the stochastic transition function. To build models, attributes such as functional compliance, reliability, vulnerability, testability, and others have been developed in combination with event-driven, finite-state machine, and state-driven models. Also presented are different types of SFS and their relationship with SDLC.

Results and Discussion. The research results include the formalization of SFS, the development of evaluation metrics, and practical recommendations for SFS analytics at all stages of SDLC, which enable proactive control of the quality, reliability, security, and compliance of software systems.

Conclusion. The formalization of the concept of SFSs, including their types, properties, and parameters, allowed for a reasonable connection to the SDLC phases. The proposed metrics and recommendations contribute to the development of SFS analytics, ensuring both the theoretical integrity of the approach and its practical applicability in the tasks of monitoring, analysis and predicting SFS. This methodology creates a new foundation for self-learning SDLC-oriented ecosystems in which SFSs are predicted, assessed and managed automatically in real-time.

Keywords: software development life cycle, software functional state, functional suitability, software state prediction, software functional state analytics, software state characteristics

© 2025 Lyashkevych M.Y. et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

The SDLC is a system model that describes the sequence of processes for creating, implementing, and maintaining a software product [1]. Each phase [2] has its own goals, artefacts, and typical risks. The structure of SDLC phases may vary depending on the software development methodology and may include some stages [2]. An advanced comparison of the software development methodologies represented in [3].

However, modern practice proves that these phases are not isolated: they are interconnected by data and knowledge flows. That is why analysis, forecasting, or continuous monitoring of SFS becomes a key condition for ensuring quality and security throughout the SDLC. The methodology determines not only the sequence of phases but also the transition mechanisms between them. It defines the methods of SFS monitoring, types of possible defects, and risk criteria. As a result, monitoring, analysis, and prediction of SFS become key mechanisms for ensuring quality, reliability, security, and providing context throughout the entire SDLC.

Due to the demand for extension to the formal SFS definition, it is difficult to establish a universal model for assessment, diagnostics, prediction, quality management, and other analytics. In most standards [4-5], especially ISO/IEC 25010:2011 [6], quality is considered as a set of characteristics such as reliability, safety, functional suitability, efficiency and others, but no mechanism for transitioning between system states is defined, which leads to a gap between the process level and the functional level and its context of the SDLC model.

Commonly, the SFS is understood as a set of parameters that describe the current behavior, performance and quality of the system relative to its objectives, resources, and environment at a given point in time [1]. This concept is key in building intelligent process monitoring systems, multi-agent SDLC control systems, and LLM-oriented systems, and its importance suggests that modern methodology must shift from a static understanding of processes to dynamic modelling of software states. However, modern quality standards, such as ISO/IEC 25010:2011, focus primarily on quality characteristics such as functional suitability, effectiveness, compatibility, reliability, security, availability, maintainability, and portability [6]. These models lack a formal definition of the system state and therefore cannot establish a general mechanism for assessing the current SFS or predicting its changes.

The architecture of modern systems is becoming increasingly complex, ranging from microservices and multi-agent systems to distributed solutions running in the cloud or hybrid environments. Such systems typically involve continuous component changes, version updates, scaling, and load balancing, thus creating a dynamic state space [7]. Also, in modern development methodologies, especially DataOps or DevSecOps, continuous monitoring and security are based on automated metrics and predictive models [8]. Without a clear state formalization, it is difficult to build systems for degradation detection, fault prediction, or adaptive recovery.

In the context of artificial intelligence and LLMs, a new type of software is emerging where a system's behavior is determined by dynamic knowledge, context, and learning outcomes. This requires describing the state not only at the code or process level, but also at the interaction level between knowledge, model, and context [9].

In recent years, scientific publications have been actively researching methods for modelling the behavior of software systems through SFS [10]. Model-based multi-objective optimization helps address complex trade-offs in software architecture quality attributes, guiding refactoring decisions and highlighting key research challenges and future directions [11].

Existing quality models, stochastic SDLC frameworks [12], and performance monitoring methods are mainly designed for manually developed software systems, or at most, for development processes supported by partial code generation tools such as code

completion assistants or systems like Copilot. In such an environment, the behavior, architecture, and quality attributes of the software or other uncertainties [13] are largely determined by human-installed solutions, while automation tools play a supporting role. Therefore, the concept of software state in these models is often simplified to a set of observable technical characteristics or performance metrics, without explicitly including a generation environment for creating software artifacts.

Modern development paradigms increasingly rely on partially or fully automated code generation, including LLMs, agent-based development pipelines, and adaptive DevSecOps workflows. In this environment, the behavior and quality of a software system are not merely the result of static design artifacts or performance metrics, but rather the result of dynamic interactions between generating models, clues, learned knowledge, configuration strategies, and the ever-changing SDLC environment. Existing stochastic SDLC models [12] and degradation or risk prediction methods do not explicitly model this context dependency and therefore lack the ability to represent software state as a function of the technical execution and generation context.

Thus, there is a methodological gap between traditional state-based quality or risk models and the requirements of generative and agent-oriented software development. Existing methods cannot provide a single state representation that can simultaneously capture execution attributes, quality and risk characteristics, contextual dependencies, and semantics of SDLC stages, while supporting probabilistic predictive modeling, interstage thinking, and state transitions.

Bridging this gap requires extending the concept of software state from static functions to context-dependent, multidimensional functional representations. Without such state representations, some important practical problems cannot be systematically solved. Especially in the early stages of the SDLC (such as requirements analysis or architecture design), when software artifacts are still partially or fully generated, it becomes impossible to reliably predict the transition from an unstable state to a failure state. Similarly, it remains inappropriate to compare different SDLC strategies or generative development processes based on expected state trajectories rather than isolated metrics. Ultimately, because recovery, regeneration, and mitigation measures depend not only on technical metrics but also on the underlying build and execution environment, it is impossible to effectively reconcile DevSecOps with stateful or intelligent development processes.

To address these issues, this paper proposes a unified abstraction of software functional states that covers the software development lifecycle phases and runtime execution, while explicitly considering context and generation dimensions. Furthermore, a probabilistic model of SFS evolution is proposed, which allows for prediction and inter-stage reasoning, as well as a state classification framework validated through Monte Carlo simulations, demonstrating the separability and robustness of SFS under uncertainty.

Therefore, the extended formalization of the SFS concept is a prerequisite for constructing the theoretical and applied foundation of software intelligent analysis and monitoring. It integrates methods such as systems analysis, artificial intelligence, ontology modelling, knowledge engineering, and other analytics. This concept lays the foundation for developing new SDLC models that ensure quality and security not only at the process level but also at the operational status level of software products with its context.

MATERIALS AND METHODS

It is important to emphasize that the proposed formal approach is not intended to replace the traditional SDLC or quality models for manually developing software systems. Instead, it extends them to modern development environments, where software artifacts are generated, in whole or in part, by automated agents, large language models (LLMs),

or hybrid human-machine workflows. In this environment, SFS cannot be fully characterized by artifact integrity or execution time metrics alone, because the fundamental determinants of system behavior are embedded in the knowledge that generates prompts and searches, configuration strategies, and the ever-changing SDLC context.

The problem of determining the SFSs is one of the key ones in theoretical computer science and software systems engineering. It is related to the fact that modern software systems are highly complex and are being characterised by high dynamics, distribution, adaptability, and context dependence. Unlike hardware systems, where the state of additional physical parameters is determined, in software, it is formed through a set of variables that describe behavior, internal resources, calculation logic, and interaction with the environment. Also, there is a lot of contextual information throughout the entire SDLC.

Existing stochastic SDLC models and degradation or risk prediction systems can successfully explain uncertainties in process execution or operational behavior. They typically assume that software artifacts are fixed or implicitly controlled by humans. Therefore, uncertainty is mainly modeled at the level of defect frequency, failure rate, or process delay. Conversely, in generative development, uncertainty permeates multiple levels, including the variability of generated code, semantic drift in requirement interpretation, and context-aware decision-making by autonomous agents. These factors exceed the representativeness of classic stochastic models, thus necessitating the introduction of context-dependent state function abstractions.

Historical approaches to “state” description

The initial conceptions of the SFS emerged within the context of automata theory, where a system was viewed as a deterministic or non-deterministic finite automaton. This model defines a system as a finite set of states: $S = \{s_1, s_2, \ldots, s_n\}$, between which discrete transitions occur under the influence of input signals or events from the input alphabet S. [14-15] Classic models, such as the Mealy machine and the Moore machine, became the fundamental mathematical basis for formally describing the behaviour of software systems as objects that react to external events and internal state changes. Harel (1987) extended this approach by proposing Statecharts, a hierarchical graphical notation for complex systems with parallelism and nested states [16]. This approach later became the basis for UML State Machine Diagrams, which are used today to model the behavior of software components [17].

In the 1990s, the problem of determining the state became practical in the V&V (verification and validation) model, where each design phase corresponds to a specific state of the system: from requirements to testing [18]. Spiral, represented in [19], viewed the state as the result of risk iteration with the advent of agile methods. The concept of state began to be seen as a dynamic context of tests, tasks, and requirements at a certain point in time [20].

The concept of “state” in computer science

In the classic definition of computer science, state refers to the collection of all stored data and context relating to the current behavior of a system at a given point in time. Officially, in [21] stated: “In computer science, the state of a program or computational system is a complete description of its current condition, including all stored information that can affect future behavior”. Therefore, a state is a snapshot of all variables, structures, and contexts that determine how the system will respond to incoming events.

For software, the “state” definition applies not only to data in memory or files, but also to the internal logical state of modules, the execution state of processes, active services, configuration settings, and component states. Therefore, SFS reflects the current state of an application, including its internal data, behavior, and readiness to perform functions.

"State" as an object of control and diagnostics

In the theory of system reliability, the concept of SFS is interpreted as the result of the interaction between the system and the environment, which determines the system's performance [22].

In modern software engineering, state is defined through a set of variables that describe the following characteristics:

$$S = \{ (q_i, r_i, c_i, e_i, t_i) \mid i = 1 \ldots n \}, \quad (1)$$

where q_i – logical state of the process, r_i – resource load, c_i – configuration parameters, e_i – external influences, t_i – time indicators.

This representation enables monitoring and prediction using state estimation, time series forecasting, and anomaly detection methods [23]. In DevOps architecture, this is achieved through an Application Performance Monitoring (APM) system that collects metrics on CPU, memory, latency, errors, and creates an instantaneous state model [24].

The relationship between "state" and the qualitative software characteristics

The standard [6] defines a set of software quality characteristics, describing the state of software in terms of functionality, efficiency, reliability, security, compatibility, availability, maintainability, and portability. A key characteristic previously associated with the concept of functional status is "Functional Suitability". This characteristic represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. This characteristic is composed of the following components [6]:

- Functional completeness – the degree to which the set of functions covers all the specified tasks and intended users' objectives.
- Functional correctness – the degree to which a product or system provides accurate results when used by intended users.
- Functional appropriateness – the degree to which the functions facilitate the accomplishment of specified tasks and objectives.

Therefore, the SFS within the ISO/IEC 25010 standard is a dynamic representation of these three components at a given point in time. A product is a good product if it can perform all the necessary functions correctly and efficiently. If some functions work partially or incorrectly, or if some functions are missing, then the product is abnormal or defective. Obviously, the quality standards assume that software quality assessment systems are static, and they capture properties at a certain point in time, but do not allow modelling the evolution of states.

Definitions in international and national standards

Functional suitability is defined as the main characteristics that describe the functional efficiency of a system under given conditions [4]. This approach permits assessment of the functional implementation degree, accuracy, and relevance, namely the parameters which make up the SFS.

In the previous version of the quality model, functional suitability was interpreted as: "The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions". In [5], the emphasis is on "capability," that is, the potential SFS in relation to its purpose.

This standard [25] introduces metrics for evaluating the characteristics of functionality, reliability, and maintainability. It effectively transforms the concept of SFS into a quantitative model through indicators, such as: "percentage of functions that perform correctly", "number of failed operations per function", "execution completeness ratio". That is, SFS can be measured numerically.

Regarding the SFSs or states when the software performs or does not perform its functions, the latest national standard [26] explains the functional suitability, which refers to the extent to which the software provides functions that meet stated and implied needs under given conditions of use.

"State" in the context of adaptive and intelligent systems

The development of adaptive systems and autonomous agents has given rise to the concept of runtime models, which describe the current state of a system and can be observed and modified during execution [27].

Kephart and Chess, in their paper [28], defined the state in the concept of autonomous computation as a set of controllable properties of a system that can be measured, compared to a reference value, and adjusted without human intervention.

In adaptive and multi-agent systems, SFSs are viewed as information projections of an agent's behavior in space, such as environment, objectives, actions, and resources [29]. This allows states to be formalized through ontology, logical predicates, or vector representations.

In the context of LLMs and generative AI systems, SFSs include not only technical parameters but also cognitive parameters such as current context, query history, word segmentation parameters, weights of inner layers, state cache, etc. [30] Bubeck et al. (2023) pointed out that the behavior of LLM is a stochastic function of current knowledge and context state, and therefore can be modeled as a stateful system [9].

In retrieval augmented generation systems and multi-agent LLM environments, "states" determine action readiness, response reliability, and contextual consistency. This allows us to view not only SFSs technically but also semantic states, thereby determining the level of cognitive consistency among agents [31].

Ukrainian scientific approaches

In the Ukrainian scientific space, the concept of SFS transcends classical automata theory and has been applied to practical problems such as risk assessment, energy modeling, ontology-based diagnostics, and context-aware monitoring in decision support systems. Furthermore, the authors of reference [12] demonstrate that the probabilistic GERT-based integration model can integrate artificial intelligence intervention into the hybrid SDLC, transforming continuous integration/continuous delivery (CI/CD) telemetry data into interpretable schedule risk predictions, thereby reducing rework cycles, shortening delivery time distribution, and supporting scenario-driven optimization under cognitive uncertainty.

In article [32], a fuzzy model of risk assessment as a function of software product state change has been formed. In that model, the concept of software state is used in its fuzzy risk assessment model. SFS is treated as a variable, and the risk level of the software product depends on this variable as a function of state changes.

The analysis of scientific publications shows that existing methods fail in providing a complete formal description of the SFS, whether process-based, behavioral, standard or cognitive ones. In most models, software state is treated as a set of internal variables or performance levels, but the factors below are not considered:

- External context of the runtime environment as infrastructure, users and security events.
- Ontological relationships between states, such as hierarchical relationships, compatibility, and transitivity.
- Evolutionary dynamics of states, such as development, degradation, and adaptation.
- Intelligent decision-making processes that influence states.

Therefore, a new formal model is required to describe the SFS as a multidimensional entity in a space of technical, behavioral, cognitive, and semantic features. Such a model

will provide a unified state representation for monitoring tasks, quality assessment, risk management, and the real-time adaptation of software systems. Thus, the SFS will represent a “snapshot” of all characteristics that determine how the system reacts to inputs, performs its functions, and maintains quality in accordance with the ISO/IEC 25010 quality characteristics such as functional suitability, reliability, efficiency, security, and maintainability.

RESULTS AND DISCUSSION

Existing stochastic software development lifecycle models mainly adopt a process-centric perspective, modeling the transitions between lifecycle stages or activities, but do not explicitly represent the internal functional states of the software system. Quality models (such as those conforming to ISO/IEC 25010) provide a static snapshot of software characteristics but do not model state evolution or contextual dependencies. Runtime monitoring and performance degradation prediction methods focus on the runtime phase and treat the Software Development Lifecycle (SDLC) environment as exogenous or fixed. In contrast, the SFS model proposed in this paper adopts a state-centric perspective, covering the SDLC phase and runtime execution, integrating quality, risk, and contextual information, and explicitly supporting prediction and probabilistic reasoning. This positioning makes the model particularly suitable for environments involving some or all of the code generation, where software behavior stems from both execution dynamics and the generation context.

Formalisation of definitions

The SFS is not only a technical execution property but a dynamic multidimensional construction combining:

- System properties such as performance, stability, reliability, and others.
- Requirement compliance, such as functional and non-functional requirements.
- Current suitability, quality, risk, and security profile.
- Operational and contextual conditions such as environment, workload, configuration, knowledge context, etc.

Software functional state is a set of parameters that reflect the current quality, operability, security, integrity, and contextual compliance of a software system at a given moment in time, relative to its functional requirements, architectural constraints, resources, and operating environment.

The SFS of the system at time *t* is defined as:

$$S(t) = \langle \phi(t), \mu(t), \sigma(t), \rho(t), \kappa(t), \tau(t) \rangle, \quad (2)$$

where: $\phi(t)$ – level of functional adequacy or requirement fulfillment; $\mu(t)$ – performance indicators such as response time, resource utilization; $\sigma(t)$ – stability indicators such as failure rate, MTBF; $\rho(t)$ – risk and security indicators such as vulnerabilities, threat probability; $\kappa(t)$ – contextual and cognitive coherence for intelligent or LLM-based systems; $\tau(t)$ – SDLC phase context such as project initiation and planning, requirement analysis, architectural design, development, testing, deployment, maintenance and operations, team and management.

Although the components of the SFS vector may appear heterogeneous—combining performance metrics, quality metrics, risk metrics, contextual consistency, and SDLC stage information – this heterogeneity reflects the fundamental nature of software states in generative and adaptive systems. From the point of view of systems theory and state assessment, the state of a system is determined not by the homogeneity of its parameters, but by the completeness of the information necessary to predict future behavior. In generative software systems, this predictive completeness cannot be achieved without

explicitly modeling the context and lifecycle dimensions. Therefore, the SFS vector is constructed as a multi-layered representation of states, including core execution states, derived quality and risk states, and contextual SDLC embeddings, which together define the evolution of the system.

The integral quality of the state is defined as:

$$Q(S) = w_\phi \phi + w_\mu \mu + w_\sigma \sigma + w_\kappa \kappa - w_\rho \rho, \quad (3)$$

where w_i are the weighting coefficients correspond to a specific system class.

A software functional state during the software development life cycle is the predicted or actual configuration of suitability, quality, risk, security, and operability parameters of a software system formed at each SDLC phase under the influence of development methodology and artefacts, team decisions, and environmental factors.

This state has a dual nature:

- Predictive (forecastive) – derived from software requirements and architectural analysis with prior distribution $\pi_0(S)$.
- Empirical (consequential) – refined by real observations during development, testing, and other stages with posterior distribution $\pi_k(S)$.

It is not enough to simply know the types of SFS (**Table 1**) during SDLC and their relationships (**Table 2**).

Table 1. **Types of SFS**

SFS name	Description	Typical indicators	Formal conditions
Operational (Normal), S_N	System functionality meets requirements	SLA met, no critical bugs, vulnerabilities	$Q(S) \geq 0.9,\ \rho < 0.1$
Degraded, S_D	Partial loss of performance or efficiency	Higher latency, resource saturation, non-critical delays	$0.7 \leq Q(S) < 0.9$
Vulnerable, S_V	Functionality with security weaknesses	CVEs, authentication flaws, data exposure	$0.6 \leq Q(S)$ $Q(S) < 0.8, \rho > 0.3$
Anomalous, S_A	System behavior deviates from expected norms	Unusual requests, unstable metrics	$metric\ deviation > 3\sigma$
Defective (Buggy), S_B	Logical or functional faults without a crash	Incorrect output, UI errors	$test\ failure,$ $but\ system\ operable$
Critical (Failure), S_F	System crash or total loss of functionality	Downtime, data loss, fatal exceptions	$Q(S) < 0.4, \sigma < 0.5$
Recovering, S_R	Recovery after failure or degradation	Rollback, restart, autoscaling	$S_F \rightarrow^{recover} S_R$
Transitional (Testing), S_T	Temporary unstable state during update or deployment	Active deployment or migration	$during\ CI/CD\ or$ $configuration\ change$
Uncertain, S_U	A scope doesn't fit expectations	Timelines mismatch, employee attrition	$Sched.perf.ind. < 1:$ $behind\ schedule$

Table 2. **Relation between SDLC phases and SFS**

SDLC phase	Possible states	Typical transition causes
Project initiation and planning	S_U , S_A , S_N	Scope definition, agreement with stakeholders
Requirement analysis	S_N , S_B , S_V	Ambiguity, inconsistency, and missing requirements
Architectural design	S_N , S_D , S_V	Architectural anti-patterns, design errors
Development	S_N , S_B , S_D	Coding errors, dependency issues
Testing	S_B , S_F , S_R	Test instability, coverage gaps
Deployment	S_T , S_R , S_N	CI/CD misconfiguration, misallocation
Maintenance and operations	S_N , S_D , S_A , S_V , S_F	Load spikes, attacks, degradation, failure
Team, management	S_U , S_N	Team composition with required skills

The model must take into account the fact that with each development phase and its corresponding stage, we are getting closer to expectations. That is, at each step of the requirements analysis, architectural design, planning, or testing, we are increasingly excluding undesirable states. The proposed SFS abstraction can be understood as a unified layer between classical quality assessment, risk modeling, and generative software engineering. In the traditional model, these aspects are analyzed independently: quality models assess consistency, risk models estimate failure probabilities, and generative mechanisms are treated as implementation tools. In contrast, SFS views quality degradation, risk escalation, and inconsistency as different manifestations of the same underlying state evolution process. This perspective enables us to reason about software behavior at various stages of the SDLC, including stages where executable artifacts are incomplete or continuously regenerated.

Dynamic state evolution models are important for modelling, analyzing, and predicting the behavior and evolution of complex systems over time. These models capture the interactions between components and the influence of external factors, thus supporting informed decision-making and effective management. Simple models do not perform both prediction and simulation, allowing researchers to predict system behavior under different conditions. By revealing internal mechanisms and feedback loops, dynamic models can enhance our understanding of complex adaptive systems.

In engineering and applied sciences, dynamic models are crucial for system design and control, ensuring system efficiency and safety. They also facilitate hypothesis testing, providing a structured approach to evaluating system responses. In fields such as economics, biology, and urban planning, dynamic models are being used as decision support tools, guiding policy or strategy development.

State transitions, in the dynamic state evolution model, during the SDLC are determined by a stochastic process:

$$S_{t+1} = \delta(S_t, A_t, E_t), \tag{4}$$

where: A_t – set of actions within SDLC phases and their stages; E_t – external factors such as requirement changes, load variations, or environment.

Transition probability is being calculated:

$$P(S_{t+1} = S_j \mid S_t = S_i, \quad A_t = a) = p_{ij}(a), \tag{5}$$

Hence, the SDLC can be represented as a graph of state transitions, forming an appropriate digital model of the development process.

As discussed above, the SFS is a complex function which has some characteristics. The target set of characteristics depends on the specific software, chosen methodology, technology stack, etc. The core characteristics of SFS with typical metrics for estimation are shown in **Table 3**.

Table 3. **Core characteristics of SFS**

Characteristic	Definition	Typical Metric
Availability	The time system remains operational	$MTBF \,/\, (MTBF \,+\, MTTR)$
Stability	Sensitivity to changes and faults	$Performance\ variance\ \sigma$
Reliability	Probability of fault-free execution	$R(t) = e^{-\lambda t}$
Security	Probability of absence of exploitable vulnerabilities	$1 - CVSS_{norm}$
Code Quality	Structural and logical correctness	$Cyclomatic\ Complexity, Coverage$
Contextual Coherence	Consistency with the environment and the knowledge base	$Semantic\ Consistency\ Index$
Risk	Probability–impact product	$R_i = P_i \cdot I_i$
Viability	Time share in healthy states	$L(S) = (T_{S_N} + T_{S_D}) \,/\, T_{total}$

The proposed set of indicators may be refined in the future depending on the objectives of the SFS study.

General formalization

SFS creates a hierarchy of system viability:

$$S_N > S_D > S_V > S_A > S_B > S_F, \tag{6}$$

where “>” indicates higher operational integrity.

Valid transitions form a stochastic graph of SDLC:

$$S_N \to S_D \to S_V \to S_F \to S_R \to S_T \to S_N, \tag{7}$$

Uncertain, anomalous, and defective states (S_U , S_A , S_B) may emerge at any SDLC phase as early indicators of potential failures.

Prior distributions of SFSs are various at different SDLC phases and their stages but can be estimated. For example, at the “Requirement analysis” phase, a prior distribution of SFS is estimated as:

$$\pi_0(S) = P(S \mid X^{(req)}), \tag{8}$$

where $X^{(req)}$ is a vector of requirement features such as completeness, security, consistency, ambiguity, etc.

As the SDLC progresses, the posterior estimate is updated using new observations O_k:

$$\pi_k(S) \propto P(O_k \mid S) \sum_{S'} P(S')\pi_{k-1}(S'). \tag{9}$$

Hence, the SFS in SDLC evolves from a probabilistic forecast to an empirically measurable condition. State Function in SDLC:

$$S_{t+1} = f\left(S_t, X^{(req)}, O_{1:t}, A_t, E_t\right), \tag{10}$$

where f describes the predictive-reactive evolution of the SFS.

Optimization criterion:

$$max_\pi E\left\{\sum_t \gamma^t [Q(S_t) - \rho(S_t) - C(A_t)]\right\}, \tag{11}$$

where policy π determines actions to maintain viable states.

SFS is a dynamic, multidimensional characteristic that describes the aggregate of quality, operability, stability, security, and contextual coherence parameters of a software system at a specific moment, formalized as a vector model S_t that evolves under the influence of internal processes and external environmental factors.

SFS in the SDLC is a stochastic function describing the sequence of transitions between SFS types:

$$\Omega = (S_N, S_D, S_V, S_A, S_B, S_F, S_R, S_T), \tag{12}$$

The statement (11) is driven by SDLC activities and artefacts – from forecastive prediction at the requirement phase to empirically validated runtime observations during operations.

Estimation of the proposed definitions

The estimation metrics of the proposed SFS and SFS during SDLC definitions should be aligned with the formal model (3) and their stochastic evolution through SDLC phases. The estimation metrics aim to:

- Quantify the current or predicted SFS.
- Detect deviations, degradation, or transitions between states.
- Support automated decision-making, diagnostics, and optimization within SDLC.
- Correlate process-level artefacts, such as requirements, commits, tests, and metrics with state-level variables.
- Detect component-level and state-level dependencies across all phases of the SDLC.
- Other software and development methodologies specifics.

The estimation dimensions represent the key measurable components of a software's functional state. They capture functional adequacy $\phi(t)$ through requirement compliance, performance $\mu(t)$ via runtime efficiency, and stability $\sigma(t)$ through reliability under perturbations. Risk and security $\rho(t)$ quantify exposure to faults or threats, while contextual coherence $\kappa(t)$ measures semantic and environmental consistency.

Finally, the lifecycle phase $\tau(t)$ situates all parameters within the SDLC process context, ensuring temporal and methodological traceability. Each SFS parameter has its own measurable indicators and a normalised score in a range of $[0, 1]$ (**Table 4**).

Composite indices for SFS estimation integrate multiple quality, performance, risk, and stability metrics into a unified score. They provide a holistic view of software health, simplify complex multidimensional assessments, enable trend detection, support automated decision-making, and allow early prediction of degradation across the SDLC.

SFS quality index $FSQI$:

$$FSQI(t) = w_\phi RC \cdot TC + w_\mu RE + w_\sigma \frac{MTBF}{MTBF_{max}} + w_\kappa SCI - w_\rho REX, \tag{13}$$

where range is $[0, 1]$.

The interpretation is:

- $FSQI \geq 0.9 \rightarrow Operational(Normal)$.
- $0.7 - 0.9 \rightarrow Degraded$.
- $< 0.7 \rightarrow Vulnerable$.

Dynamic risk index DRI:

$$DRI(t) = \sum_{i=1}^{k} P_i \cdot I_i \cdot w_i, \tag{14}$$

where P_i – probability of event i, I_i – impact magnitude, w_i – weight.

The interpretation is:

- $DRI > 0.4 \rightarrow risk\ accumulation\ trend, requiring\ mitigation\ actions$.

SFS stability index $FSSI$:

$$FSSI(t) = 1 - \frac{\sigma_{perf}(t) + \sigma_{error}(t)}{\sigma_{max}}. \tag{15}$$

The interpretation is:

- $Higher\ values = better\ stability$.
- $Values < 0.5\ indicate\ transition\ toward\ S_D\ or\ S_F$.

SDLC functional integrity LFI, which measures how consistently the system remains in acceptable states across SDLC phases:

$$LFI(t) = \frac{1}{T_{total}} \sum_{phases} Q(S_{phase}) \cdot \Delta t_{phases}. \tag{16}$$

Composite SDLC state metric $CLSM$ allows for end-to-end lifecycle monitoring:

Table 4. **Quantitative Metrics Calculations**

Dimension	Metric	Calculation	Range	Interpretation
Functional adequacy, $\phi(t)$	Requirement Coverage, RC	$RC = \frac{N_{implemented}}{N_{specified}}$	$[0,1]$	Degree of implemented requirements
	Requirement Consistency, $RCon$	$RCon$ = NLP-based coherence score between requirements	$[0,1]$	Higher = fewer conflicts
	Test Pass Ratio, TPR	$TPR = \frac{N_{passed}}{N_{total}}$	$[0,1]$	Functional correctness
	Technical Coherence, TC	$TC = 1 / \frac{N_{adapted}}{N_{expected}}$	$[0,1]$	The right tech stack, the higher = the fewer mismatches
Perform., $\mu(t)$	Response Efficiency, RE	$RE = 1 - \frac{t_{resp}}{t_{SLA}}$	$[0,1]$	Lower latency → higher RE
	Resource Utilization Efficiency, RUE	$RE = 1 - \frac{CPU + MEM}{100}$	$[0,1]$	Optimal performance balance
Stability, $\sigma(t)$	Mean Time Between Failures, $MTBF$	$MTBF$ = Empirical	$[0,\infty)$	Higher = more stable
	Variance of Key Metrics, σ_{perf}	σ_{perf} = standard deviation of latency/load	norm. $[0,1]$	Lower = more stable
Security/ risk, $\rho(t)$	Vulnerability Index, VI	$VI = 1 - \frac{\Sigma(CVSS_i \cdot w_i)}{\Sigma w_i \cdot 10}$	$[0,1]$	1 = no vulnerabilities
	Risk Exposure, REX	$REX = P(Event) \times Impact$	$[0,1]$	Expected severity of failure
	Attack Surface Ratio, ASR	$ASR = \frac{N_{exposed\ interfaces}}{N_{total\ interfaces}}$	$[0,1]$	Lower = more secure
Contextual coherence, $\kappa(t)$	Semantic Consistency Index, SCI	$SCI = sim(v1, v2)$	$[0,1]$	High value = coherent with context
	Configuration Drift, CD	$CD = 1 - sim(conf_{runtime}, conf_{baseline})$	$[0,1]$	0 = perfect match
Lifecycle robustness, $\tau(t)$	Phase Consistency, PC	Entropy of transitions $PC = -\Sigma\, p_i \log p_i$	$[0, logN]$	Lower = more deterministic SDLC
	Temporal Integrity, TI	TI = delay between planned and actual phase completion	norm. $[0,1]$	Schedule stability

$$CLSM(t) = \frac{1}{N_{phases}} \sum_{k=1}^{N_{phases}} (FSQI_k - DRI_k), \quad (17)$$

where range is $[0, 1]$.

The interpretation is:

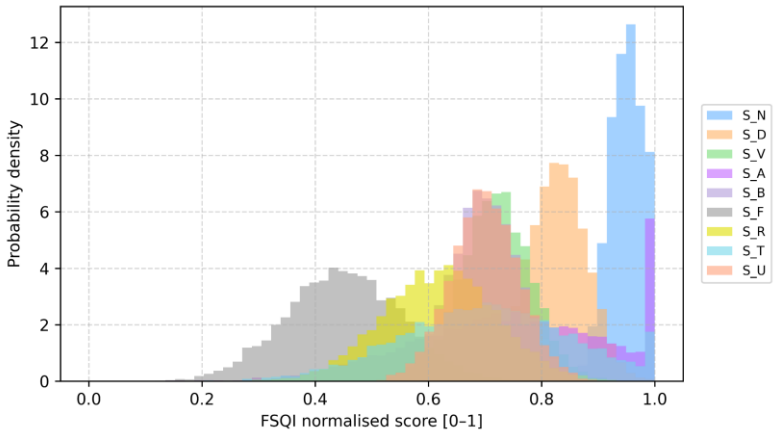
- $CLSM > 0.7 \rightarrow SDLC\ is\ healthy\ (high\ functional\ integrity)$.
- $CLSM\ between\ 0.4 - 0.7 \rightarrow partial\ degradation$.
- $CLSM < 0.4 \rightarrow instability\ or\ risk\ accumulation$.

At the moment, we are setting the thresholds for SFS classification that are defined by analyzing normalized quality, risk, and stability metrics such as $FSQI$, DRI and $FSSI$across multiple SDLC phases and mapping their statistical distributions to empirically observed states. The entire table with the thresholds is shown in **Table 5**.

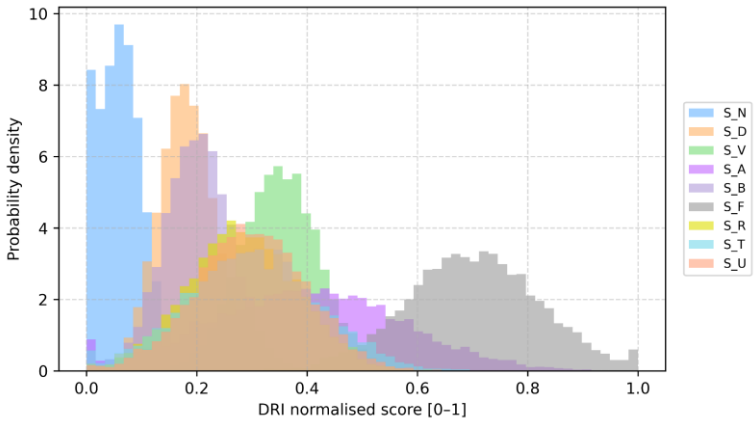
Table 5. **Thresholds for SFS classification**

SFS	$FSQI$ range	DRI range	$FSSI$ range	Qualitative interpretation
Normal, S_N	0.90 – 1.00	< 0.10	> 0.85	Fully stable, performant, secure; system meets or exceeds all requirements.
Degraded, S_D	0.78 – 0.90	0.10 – 0.22	0.70 – 0.85	Minor performance decline or partial overload without failure; functionality intact.
Vulnerable, S_V	0.66 – 0.78	0.22 – 0.38	0.60 – 0.78	The system is functional, but operates under elevated security or reliability risk.
Anomalous, S_A	variable (≈ 0.55–0.85)	spikes > 0.45 (> 3σ events)	0.45 – 0.70 (fluctuating)	Behavioral deviations from expected norms; potential precursor to defect or attack.
Defective, S_B	0.60 – 0.75 (local)	< 0.25	0.55 – 0.75	Logical or functional errors appear without a full crash, incorrect outputs.
Failure, S_F	< 0.55	> 0.45	< 0.45	Critical system disruption, downtime, or data loss requires immediate recovery.
Recovering, S_R	0.58 – 0.72 (transient ↑)	decreasing 0.30 → 0.20	increasing 0.55 → 0.75	System restoring from failure toward normal; self-healing or restart in progress.
Transitional, S_T	0.60 – 0.80 (transient)	0.20 – 0.35	0.55 – 0.70	Short-term unstable configuration during deployment, migration, or CI/CD.
Uncertain, S_U	0.68 – 0.76	0.25 – 0.35	0.60 – 0.70	State of scope misalignment, ambiguous requirements, or resource/timeline mismatch.

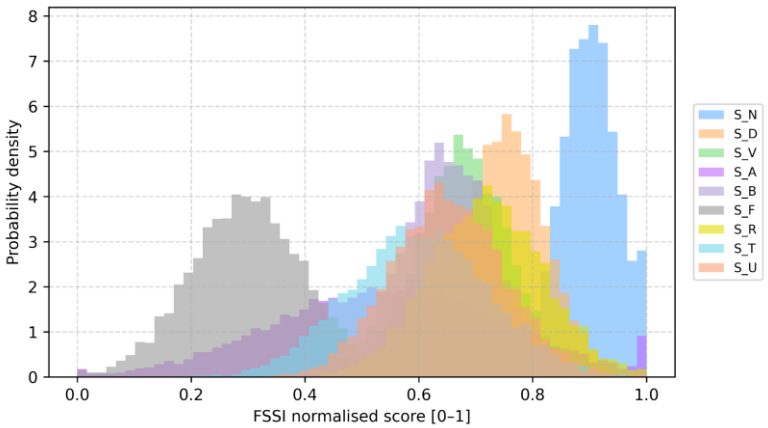
To prove the formalization, we currently validate thresholds via dynamic modelling and Monte Carlo simulations of SDLC (**Fig. 1**), comparing predicted and actual state transitions, ensuring consistent separability and stability of states across iterations and empirical test datasets.



(a)



(b)



(c)

Fig. 1. Results of Monte Carlo simulation for $FSQI$ (a); DRI (b) and $FSSI$ (c).

The generated distributions of $FSQI$, DRI and $FSSI$ across the nine functional software states: S_N (marked as 'S_N' in **Fig. 1**), $S_D, S_V, S_A, S_B, S_F, S_R, S_T, S_U$) demonstrate that the proposed mathematical formalization reliably produces distinct, separable, and meaningful clusters that reflect the expected behavior of software under varying levels of quality, risk, and stability. Each distribution captures the theoretical assumptions of the model while revealing realistic overlaps in transitional or ambiguous operational conditions.

The simulations correctly reproduce the expected real-world fuzziness of these states. It has shown that no single metric is sufficient, but the vectors together produce reliable decision boundaries.

The results achieved provide an effective and applicable baseline for application in SFS analytics, real-time SDLC monitoring, and decision support. Monte Carlo modelling and updated classification thresholds can reliably distinguish SFS through measurable indicators $FSQI$, DRI and $FSSI$ thereby enabling continuous assessment of quality, risk, and stability. These models can be integrated into CI/CD pipelines, DevSecOps control panels, and as digital twin models of SDLC to predict state transitions, detect early performance degradation, and automatically execute recovery strategies.

From the perspective of LLM-based and agentic SDLCs, the concept of SFS takes on additional meaning. In these systems, software behavior is influenced by nondeterministic generative processes, evolving internal representations, and external knowledge sources. Therefore, SFS should not only include execution-level attributes, but also semantic consistency, contextual validity, and consistency between the produced artifacts and the software development lifecycle goals. The proposed SFS model provides a formal basis for representing these dimensions without binding the method to a specific generative technique.

CONCLUSION

This paper proposes a formal and computational framework for modeling the SFS throughout the SDLC, extending classic state-based and quality-oriented approaches to context-aware and probabilistic representations.

The results show that traditional models are insufficient to cope with new development paradigms that involve some or all of the code generation, in which software behavior depends on execution technology and the production environment.

The proposed SFS abstraction enables predictive analysis, interphase thinking, and state-aware orchestration of DevSecOps and agent-based development pipelines, providing a foundation for intelligent SDLC monitoring and decision support.

The next step is to transform the theoretical model into a working platform for the implementation of the proposed model in real-world SDLCs, integrating it with generative processes, and evaluating its effectiveness in large-scale industrial environments.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [M.L.]; methodology, [M.L.]; validation, [R.S., V.L.]; formal analysis, [M.L.]; investigation, [M.L.]; resources, [M.L.]; data curation, [M.L.]; writing – original draft preparation, [M.L.]; writing – review and editing, [V.L.]; visualization, [M.L.]; supervision, [R.S., V.L.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Lyashkevych, M. Y., Lyashkevych, V. Y., & Shuvar, R. Y. (2025). Security and other risks related to LLM-based software development. Ukrainian Journal of Information Technology, 7(1), 86–96. https://doi.org/10.23939/ujit2025.01.086.

[2] Lyashkevych, M. Y., Rohatskiy, I. Y., Lyashkevych, V. Y., & Shuvar, R. Y. (2024). Software risk taxonomy creation based on the comprehensive development process. Science and Technology: New Horizons of Development 209 – Electronics and Information Technologies, 1(27), 59–71. https://doi.org/10.30970/eli.27.5.

[3] Hossain, Mohammad. (2023). Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management. International Journal For Multidisciplinary Research. https://doi.org/10.36948/ijfmr.2023.v05i05.6223.

[4] State Enterprise “UkrNDNC.” (2016). DSTU ISO/IEC 25010:2016 – Systems and software engineering – Systems and software quality requirements and evaluation – System and software quality model. Kyiv, Ukraine: SE “UkrNDNC.” (In Ukrainian; translated title.)

[5] State Committee of Ukraine for Technical Regulation and Consumer Policy. (2005). DSTU ISO/IEC 9126-1:2005 – Information technology – Software product quality – Part 1: Quality model. Kyiv, Ukraine: Derzhspozhyvstandart Ukrainy. (In Ukrainian; translated title.)

[6] International Organization for Standardization. (2011). ISO/IEC 25010:2011 – Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva, Switzerland: ISO. URL: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010.

[7] Jamshidi, P., Pahl, C., Lewis, J., & Tilkov, S. (2020). Microservices: The journey so far and challenges ahead. IEEE Software, 38(1), 24–31. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8354433.

[8] R. Yedida and T. Menzies, "How to Improve Deep Learning for Software Analytics (a case study with code smell detection)," 2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR), Pittsburgh, PA, USA, 2022, pp. 156-166, doi: https://doi.org/10.1145/3524842.3528458.

[9] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Li, Y., Lundberg, S., Nori, H., & others. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv preprint arXiv:2303.12712. https://doi.org/10.48550/arXiv.2303.12712.

[10] S. Silva, A. Tuyishime, T. Santilli, P. Pelliccione and L. Iovino, "Quality Metrics in Software Architecture," 2023 IEEE 20th International Conference on Software Architecture (ICSA), L'Aquila, Italy, 2023, pp. 58-69, doi: https://doi.org/10.1109/ICSA56044.2023.00014.

[11] D. Di Pompeo and M. Tucci, "Quality Attributes Optimization of Software Architecture: Research Challenges and Directions," 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), L'Aquila, Italy, 2023, pp. 252-255, doi: https://doi.org/10.1109/ICSA-C57050.2023.00061.

[12] Semenov, S., Tsukur, V., Molokanova, V., Muchacki, M., Litawa, G., Mozhaiev, M., & Petrovska, I. (2025). Mathematical Model of the Software Development Process with Hybrid Management Elements. Applied Sciences, 15(21), 11667. https://doi.org/10.3390/app152111667.
[13] Li, Can & Grossmann, Ignacio. (2021). A Review of Stochastic Programming Methods for Optimization of Process Systems Under Uncertainty. Frontiers in Chemical Engineering. 2. 622241. https://doi.org/10.3389/fceng.2020.622241.
[14] Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Introduction to automata theory, languages, and computation (3rd ed.). Pearson/Addison Wesley.
[15] Lewis, H. R., & Papadimitriou, C. H. (1998). Elements of the theory of computation (2nd ed.). Prentice Hall.
[16] Harel, D. (1987). Statecharts: A visual formalism for complex systems. Science of Computer Programming, 8(3), 231–274. https://doi.org/10.1016/0167-6423(87)90035-9.
[17] OMG (Object Management Group). (2017). Unified modeling language (UML) specification (Version 2.5.1, OMG Formal Document No. 17-12-01). https://www.omg.org/spec/UML/2.5.1/.
[18] Forsberg, K., Mooz, H., & Cotterman, H. (2005). Visualizing project management: Models and frameworks for mastering complex systems (3rd ed.). Wiley. https://www.wiley.com/en-us/Visualizing+Project+Management%3A+Models+and+Frameworks+for+Mastering+Complex+Systems%2C+3rd+Edition-p-x000260487.
[19] Boehm, B. (1988). A spiral model of software development and enhancement. Computer, 21(5), 61–72. https://doi.org/10.1109/2.59.
[20] Beck, K. (2005). Extreme programming explained: Embrace change (2nd ed.). Addison-Wesley.
[21] State (computer science). (2024). Wikipedia, The Free Encyclopedia. URL: https://en.wikipedia.org/wiki/State_(computer_science).
[22] Musa, J. D. (1998). Software reliability engineering: More reliable software, faster and cheaper. McGraw-Hill.
[23] Menzies, T., & Zimmermann, T. (2023). Software analytics in DevOps. IEEE Transactions on Software Engineering, 49(3), 512–530. https://doi.org/10.1109/TSE.2022.3175113.
[24] Kim, G., Debois, P., Willis, J., & Humble, J. (2021). The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations (2nd ed.). IT Revolution Press.
[25] International Organization for Standardization. (2011). ISO/IEC 25010:2011 — Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. ISO. https://www.iso.org/standard/35746.html.
[26] DSTU ISO/IEC 25010:2025. (2025). Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — System and software quality model (ISO/IEC 25010:2023, IDT). Kyiv: SE “UkrNDNC”. https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=116491.
[27] Cheng, B. H. C., de Lemos, R., Giese, H., Müller, H., Shaw, M., & Uchitel, S. (Eds.). (2009). Software engineering for self-adaptive systems (Vol. 5525). Springer. https://doi.org/10.1007/978-3-642-02171-8.
[28] Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. Computer, 36(1), 41–50. https://doi.org/10.1109/mc.2003.1160055.
[29] Wooldridge, M. (2002). An introduction to multiagent systems. John Wiley & Sons. https://uranos.ch/research/references/Wooldridge_2001/TLTK.pdf.

[30] Bommasani, R., Hudson, D. A., Adeli, E., Agrawal, P., Ahuja, S., Argyriou, A., ... Liang, P. (2022). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258. http://arxiv.org/abs/2108.07258.
[31] Tran, Khanh-Tung & Dao, Dung & Nguyen, Minh-Duong & Pham, Viet & O'Sullivan, Barry & Nguyen, Hoang. (2025). Multi-Agent Collaboration Mechanisms: A Survey of LLMs. 10.48550/arXiv.2501.06322. https://doi.org/10.48550/arXiv.2501.06322.
[32] Pomorova, O. Fuzzy system of the evaluation and prediction of overall risks in software development [Text] / O. Pomorova, M. Lyashkevych //Proceedings of the 6th International Conference ACSN-2013. – Lviv: Ukraine Technology, 2013. – Pp.126-129.

ВИЗНАЧЕННЯ ТА ФОРМАЛІЗАЦІЯ КОНЦЕПЦІЇ ФУНКЦІОНАЛЬНОГО СТАНУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОТЯГОМ УСЬОГО ЖИТТЄВОГО ЦИКЛУ РОЗРОБКИ

Марія Ляшкевич**, *Василь Ляшкевич*, *Шувар Роман
Кафедра системного проектування
Факультет електроніки та комп'ютерних технологій
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005, Львів, Україна

АНОТАЦІЯ

Вступ. У сучасному світі програмне забезпечення (ПЗ) є критично важливим компонентом будь-якої інформаційної системи. Його розробка вимагає значних ресурсів і складних технічних рішень, а розвиток технологій настільки швидкий, що не всі поняття та визначення в галузі ПЗ є чітко формалізованими. Це особливо стосується функціонального стану ПЗ (ФСПЗ) упродовж життєвого циклу розробки (SDLC), адже передбачити всі можливі ФСПЗ практично неможливо через динамічність середовищ, зміни вимог, взаємодію компонентів і поведінку учасників проєкту. Це створює виклик для формалізації, аналізу, прогнозу та моніторингу та управління цими станами.

Матеріали та методи. Визначення та формалізації ФСПЗ охоплюють концепції із теорії станів у комп'ютерних науках, моделі якості з міжнародних стандартів ISO/IEC 25010:2011 та ДСТУ ISO/IEC 9126-1:2005. На основі цих джерел реалізуються формалізовані підходи до опису та відстеження змін у ФСПЗ протягом SDLC. Визначені поняття ФСПЗ та ФСПЗ протягом SDLC формалізовані математично, що дозволяє будувати динамічні моделі еволюції станів протягом SDLC на основі стохастичної функції переходів. Для побудови моделей використовуються атрибути якості такі як функціональна відповідність, надійність, вразливість, придатність до тестування та інші у комбінації з подієво-орієнтованими, автоматними та стано-орієнтованими моделями. Також наведено різні типи ФСПЗ та їх зв'язок із SDLC.

Результати. Отримані результати дослідження охоплюють формалізацію ФСПЗ, розробку оціночних метрик та практичні рекомендації для аналітики станів ПЗ на всіх етапах SDLC, що дозволяє забезпечити проактивний контроль якості, надійності, безпеки та відповідності програмних систем.

Висновки. Формалізація поняття функціональних станів ПЗ, включно з їх типами, властивостями та параметрами, дозволила встановити обґрунтований зв'язок із фазами SDLC. Запропоновані метрики та рекомендації сприяють розвитку аналітики станів програмних систем, забезпечуючи як теоретичну цілісність підходу, так і його

практичну застосовність у завданнях моніторингу, аналізу та прогнозування стану ПЗ. Ця методологія створює нову основу для самонавчальних SDLC-орієнтованих екосистем, у яких ФСПЗ прогнозуються, оцінюються та керуються автоматично в реальному часі.

Ключові слова: життєвий цикл програмного забезпечення, функціональний стан програмного забезпечення, функціональна придатність, прогнозування стану програмного забезпечення, аналітика функціонального стану програмного забезпечення, характеристики стану програмного забезпечення

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
14 November, 2025	01 December, 2026	01 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 171–178
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 621.317.33

THE ELECTRICAL IMPEDANCE OF A METAL-SILICON STRUCTURE WITH A THIN SiO_2 LAYER

Dmytro Slobodzyan**, *Markiyan Kushlyk*, *Yaroslav Shpotyuk*, *Roman Lys*, *Andriy Luchechko
Department of Sensor and Semiconductor Electronics,
Ivan Franko National University of Lviv,
107 Tarnavskoho Str., 79017 Lviv, Ukraine

Slobodzyan, D., Kushlyk, M., Shpotyuk, Ya., Lys, R., Luchechko, A. (2025). The Electrical Impedance of a Metal-Silicon Structure with a Thin SiO_2 Layer. *Electronics and Information Technologies*, 32, 171–178. https://doi.org/10.30970/eli.32.12

ABSTRACT

Background. Silicon-based Metal-Oxide-Semiconductor (MOS) structures with an SiO_2 dielectric layer are widely used in various modern electronic devices. Therefore, investigating the electrical properties of such structures remains a relevant task. Impedance spectroscopy is particularly valuable for studying these properties, as it enables the examination of charge transport mechanisms, interfacial properties, and the determination of parameters for equivalent electrical circuits.

Materials and Methods. Samples of n-type monocrystalline silicon were investigated, featuring the following characteristics: doping impurity – arsenic, resistivity – ρ = 0.003 Ohm·cm, thickness – 0.5 mm, area – 30 mm^2. A metal-semiconductor-metal type structure (Ag-SiO_2-Si-SiO_2-Ag) was formed on the surface of such a sample.

To study the frequency dependencies of the impedance of the formed structure, a setup based on the HIOKI IM3536 RLC meter was used.

Results and Discussion. The frequency dependencies of the real and imaginary components of the impedance of the Ag-SiO_2-Si-SiO_2-Ag structure were investigated in the frequency range from 4 Hz to 8 MHz at different excitation electric field amplitudes. It was established that for all values of the excitation signal amplitude, the real component of the impedance, *Re(Z)*, decreases as the frequency and AC excitation amplitude increase. The frequency dependencies of the imaginary component of the impedance, *Im(Z)*, demonstrate the presence of relaxation maxima. With an increase in the excitation signal amplitude, the amplitude of these maxima decreases significantly. The observed changes in the impedance of such a structure can be explained based on its multilayer nature and the properties of highly-doped silicon and the thin layers of native silicon oxide.

Conclusion. The impedance spectroscopy study of Ag-SiO_2-Si-SiO_2-Ag structures with native oxide layers revealed a strong non-linear dependence of electrical properties on the AC signal amplitude. The results demonstrate that in MOS structures with ultra-thin oxides, the “insulating” layer acts as a field-dependent tunneling resistor. Impedance spectroscopy proves to be an effective method for distinguishing between capacitive accumulation and tunneling leakage regimes in such highly doped semiconductor systems.

Keywords: silicon, impedance spectroscopy, MOS structure, Nyquist plot.

INTRODUCTION

Investigating the electrical properties of Metal-Oxide-Semiconductor (MOS) structures is a highly relevant task in modern electronics. Such structures form the basis for a wide range of devices, including memory elements, sensors, photovoltaic converters, and other

© 2025 Dmytro Slobodzyan et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

components of integrated circuits [1-3]. Systems containing thin dielectric layers, such as native silicon oxide, and highly-doped semiconductor substrates are of particular interest. The properties of such structures are largely determined by the characteristics of the dielectric layers and the dielectric-semiconductor and metal-dielectric interfaces, which necessitate the use of comprehensive methodologies for their study.

Impedance spectroscopy is particularly valuable for studying these properties [4-6]. This technique allows for the investigation of charge transport mechanisms, polarization phenomena, interfacial properties, and the determination of parameters for equivalent electrical circuits. In the case of structures with ultra-thin dielectric layers (on the order of several nanometers) and highly doped semiconductors, such as n^+-Si, quantum-mechanical effects, particularly the tunneling of charge carriers through the dielectric barrier, may play a significant role [7, 8]. Understanding and controlling these processes are critically important for the further development and miniaturization of electronic devices.

MATERIALS AND METHODS

In this work, a sample of monocrystalline silicon doped with an arsenic donor impurity was used. The resistivity of the working surface with a (111) crystallographic orientation was ρ = 0.003 Ohm·cm. The surface area of the experimental sample was 30 mm^2, and the thickness was 0.5 mm. The surface treatment was carried out using isopropyl alcohol to remove physisorbed molecules. However, a native oxide (SiO_2) remained on the silicon surface, the thickness of which, according to experimental studies, can be several nanometers [9].

The next stage involved forming the metal contact by applying a conductive paste containing silver particles to both planes, followed by curing in air at room temperature. Simultaneously, silver wires were attached to the film at the moment of application to serve as lead contacts. Silver is often used in integrated electronics for the fabrication of silicon structures and metallization in general, as it is resistant to oxidation and corrosion and is a less expensive raw material compared to other noble metals [10].

The study of the frequency dependencies of the impedance of the formed structure was performed on an experimental setup using the HIOKI IM3536 RLC meter. During the experiment, the real *Re*(*Z*) and imaginary *Im(Z)* components of the impedance were measured in the frequency range from 4 Hz to 8 MHz. The amplitude of the alternating electrical signal was 0.1 V, 0.5 V, and 1.0 V. All measurements were carried out at room temperature.

RESULTS AND DISCUSSION

Fig. 1 presents the frequency dependencies of the Real (a) and Imaginary (b) components of the impedance for the Ag-SiO_2-Si-SiO_2-Ag structure at different excitation electric field amplitudes: U = 0.1 V, 0.5 V, and 1.0 V.

It is seen from **Fig. 1a** that for all values of the excitation signal amplitude, the Real component of the impedance, *Re(Z)*, decreases with increasing frequency. A significant decrease in *Re(Z)* is also observed with an increase in the magnitude of the applied voltage over the entire investigated frequency range. At low frequencies (around 10 Hz), *Re(Z)* decreases by a factor of 13 (from approximately 2600 Ohm at U = 0.1 V to approximately 200 Ohm at U = 1.0 V. The frequency dependencies of the Imaginary component of impedance (*Im(Z)*), shown in **Fig. 1b**, demonstrate the presence of relaxation maxima. With an increase in the applied voltage, the amplitude of these maxima decreases by more than 11 times (from ~ 1100 Ohm at U = 0.1 V to ~ 95 Ohm at U = 1.0 V), and their position shifts toward the higher frequency region (from ~ 10 kHz at U = 0.1 V to ~ 50-70 kHz at U = 1.0 V).

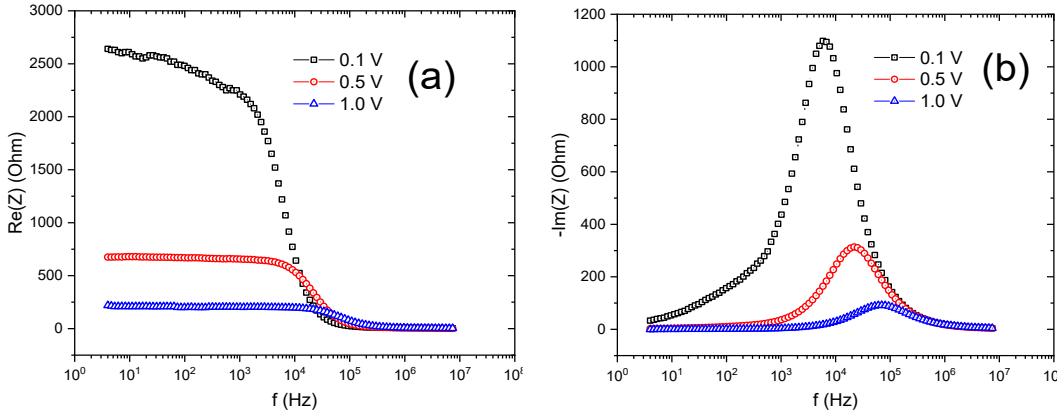


Fig. 1. Frequency dependencies of the Real (*Re(Z)*) (a) and Imaginary (*Im(Z)*) (b) components of the impedance of the Ag-SiO_2-Si-SiO_2-Ag structure at different excitation electric field amplitudes.

For an analysis of the impedance characteristics, impedance hodographs (Nyquist plots) – the dependencies of *Im(Z)* on *Re(Z)* – were plotted. These diagrams are presented in **Fig. 2**. A dominant, somewhat distorted semicircle is observed on all Nyquist plots. The high-frequency intercept of the plot with the *Re(Z)* axis is close to zero for all voltages. The most noticeable feature is the significant decrease in the diameter of this semicircle with an increase in the amplitude of the electric field. At U = 0.1 V, the diameter of the semicircle is approximately 2600 Ohm. When the voltage is increased to U = 0.5 V, the diameter decreases to approximately 650-700 Ohm, and at U = 1.0 V, to approximately 200 Ohm.

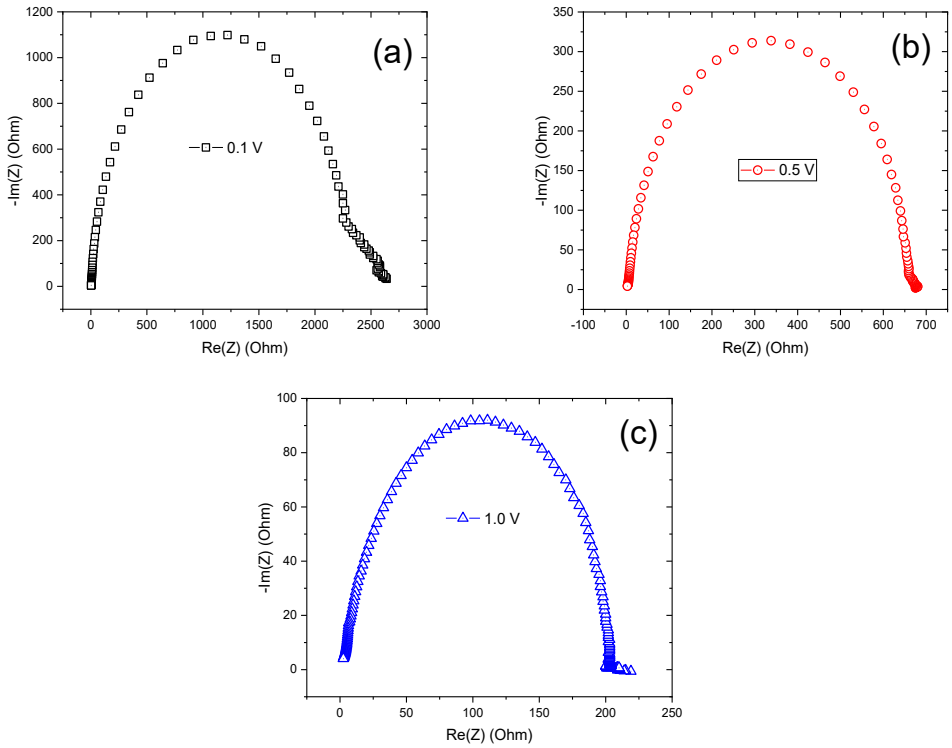


Fig. 2. Impedance hodographs (Nyquist plots) of the Ag-SiO_2-Si-SiO_2-Ag structure at excitation electric field amplitudes of U = 0.1 V (a), 0.5 V (b), and 1.0 V (c).

The observed changes in the impedance of the investigated Ag-SiO_2-Si-SiO_2-Ag structure can be explained based on the physical processes in the highly-doped silicon, the thin layers of native oxide, and the interface under the influence of the electric field.

The small value of the high-frequency series resistance, R_S (the high-frequency intercept of the hodograph with the *Re(Z)* axis), is attributed to the low bulk resistance of the highly-doped n+-silicon (resistivity ρ = 0.003 Ohm·cm), the resistance of the Ag metal contacts, and the connecting wires. The relative stability of R_S under different AC excitation amplitudes is expected, since it is primarily determined by the bulk resistance of the highly-doped silicon substrate and the ohmic metallic contacts. Due to the high carrier concentration in the n^+-Si, the bulk conductivity remains constant and is not significantly modulated by the applied electric field in the studied voltage range, unlike the barrier resistance.

The presence of a single dominant semicircle on the Nyquist plot may indicate a single primary relaxation process in the investigated frequency range. This semicircle can be associated with a parallel connection of a resistance (R_P) and a capacitive element, which reflects the charge transport efficiency (tunneling resistance) and the dielectric charge storage capability (oxide capacitance) of the ultrathin SiO_2 layers and the Si(n^+)-SiO_2 interfaces.

The key feature of the obtained results is the strong dependence of the semicircle diameter, R_P (which corresponds to the low-frequency value of *Re(Z)* minus R_S), on the applied excitation signal amplitude. The significant decrease in R_P (from ~ 2600 Ohm at U = 0.1 V to ~ 200 Ohm at U = 1.0 V) indicates a substantial increase in the efficiency of charge transport through the Ag-SiO_2-Si structure. Considering the use of highly-doped n^+-silicon (high carrier concentration) and the presence of very thin native SiO_2 oxide layers (usually several nanometers), the most probable mechanism explaining such behavior is the quantum-mechanical tunneling of charge carriers through the SiO_2 dielectric barriers [7, 8]. With an increase in the applied voltage, the electric field in the oxide layers increases, which leads to an exponential increase in the probability of electron tunneling (e.g., by direct tunneling or Fowler-Nordheim tunneling mechanisms). This effectively reduces the resistance of the dielectric layers. Although the applied voltage also modulates the Space-Charge Region (SCR) in silicon at the SiO_2 interface, for highly-doped n^+-Si, this region is extremely thin, and its modulation likely cannot significantly impact the large changes in R_P compared to the tunneling effect [8].

The decrease in the amplitude of the maxima on the frequency dependencies of *Im(Z)* (**Fig. 1b**) and their shift toward the higher frequency region with increasing AC excitation signal amplitude also aligns with the described model. The relaxation frequency, f_{PEAK}, which corresponds to the apex of the semicircle on the Nyquist plot or the maximum on the *Im(Z)* curve, is approximately determined by $f_{PEAK}/(2\pi \cdot R_P \cdot C_{EFF})$, where C_{EFF} is the effective capacitance of the structure (mainly determined by the capacitance of the SiO_2 layers). Since R_P sharply decreases with increasing voltage, while C_{EFF} changes much less, f_{PEAK} increases. The decrease in the *Im(Z)* peak height is also a consequence of the decrease in R_P. According to the interpretation proposed above, the total impedance of the Ag-SiO_2-Si-SiO_2-Ag structure will consist of the interface impedance and the bulk silicon resistance:

$$Z_{total}(\omega) = Z_{int}(\omega) + R_{Si}, \quad (1)$$

where $Z_{int}(\omega)$ is the complex interface impedance, and R_{Si} is the real-valued bulk resistance of silicon.

The interface impedance ($Z_{int}(\omega)$) is the most complex component, as it describes the characteristics of the Ag-SiO_2-Si(n^+) structure. The presence of a thin SiO_2 dielectric

layer introduces capacitive and tunneling effects, which are particularly noticeable during AC impedance measurements [11]. The interface impedance model (Z_{int}) is represented as a parallel combination of elements that model physical processes such as the oxide layer capacitance (C_{OX}) and the tunneling resistance through the oxide (R_{tun}), and a series combination of the SCR (Space-Charge Region) impedance element in silicon (Z_{SC}). Let's consider this in more detail.

The capacitance of the thin SiO_2 layer is defined as:

$$C_{OX} = \frac{\varepsilon_{OX}\varepsilon_0 S}{d_{OX}}, \tag{2}$$

ε_{OX} – the dielectric permittivity of SiO_2 (~ 3.9), ε_0 – the electric constant, S – the contact area, d_{OX} – the oxide thickness (a few nm). Due to the small thickness of d_{OX}, this capacitance will be quite significant [12, 13].

Since the SiO_2 layer is very thin (several nanometers), electrons can tunnel through it [14]. This process provides a conductive path characterized by a tunneling resistance (R_{tun}), which is effectively parallel to C_{OX}. It is the presence of this tunneling current that ensures the DC conductivity of the contact. As established in theory and confirmed by our voltage-dependence results, R_{tun} is strongly dependent on the electric field, oxide thickness, and barrier height.

An SCR is formed in the silicon under the oxide. For n^+-silicon, this region will be very thin. Depending on the applied voltage and the metal work function, the silicon under the oxide can be in a state of accumulation or weak depletion. In n^+-silicon, where the carrier concentration is very high, the SCR capacitance C_{SC} is usually very large (especially in the accumulation regime). For the studied silicon with ρ = 0.003 Ohm·cm (corresponding to a doping concentration of $N \approx 10^{19}$ cm^{-3}), the estimated bulk and SCR resistance is less than 10^{-3} Ohm, which is negligible compared to the interface resistance (10^2 - 10^3 Ohm). Similarly, due to the extremely thin depletion width in degenerate semiconductors, the SCR capacitance C_{SC} is sufficiently large compared to the oxide capacitance C_{OX}, making the contribution of the SCR to the total impedance negligible.

Since the capacitance C_{SC} is very large and the resistance R_{SC} is small, the SCR impedance (Z_{SC}) can be negligible compared to the impedance of the oxide layer (Z_{OX}), especially at non-high frequencies. In this case, the interface impedance can be approximately described only by the oxide components:

$$Z_{int}(\omega) \approx Z_{OX}(\omega) = \left(\frac{1}{R_{tun}} + j\omega C_{OX}\right)^{-1}. \tag{3}$$

Then the impedance model is defined as:

$$Z_{total}(\omega) = R_{Si} + \frac{R_{tun}}{1 + j\omega R_{tun} C_{OX}}. \tag{4}$$

Thus, according to the physical interpretation, the equivalent circuit of the impedance is a series connection of a resistor with a parallel *RC* circuit.

However, the experimental results of impedance hodographs of the Ag-SiO_2-Si-SiO_2-Ag structure (**Fig. 2**) show a slightly deformed (depressed) character of the semicircles, indicating the necessity of replacing the capacitor with a Constant Phase Element (CPE) and requiring consideration of more complex models to explain the obtained results [15]. Detailed equivalent circuit modeling and parameter determination will be the subject of a future study.

CONCLUSION

The impedance spectroscopy study of Ag-SiO_2-Si-SiO_2-Ag structures with native oxide layers revealed a strong non-linear dependence of electrical properties on the AC signal amplitude. The following key results were obtained.

Increasing the excitation signal amplitude from 0.1 V to 1.0 V leads to a drastic reduction in the real component of impedance (*Re(Z)*) by more than 10 times. This confirms that charge transport is governed by field-enhanced tunneling through the ultra-thin SiO_2 barriers rather than by simple dielectric polarization.

The imaginary impedance component (*Im(Z)*) exhibits relaxation maxima that decrease in amplitude and shift toward higher frequencies with increasing electric field. This behavior indicates a reduction in the effective parallel resistance of the structure, consistent with the tunneling model.

The Nyquist plots appear as depressed semicircles, indicating a deviation from ideal Debye relaxation. This suggests a distribution of relaxation times caused by the inhomogeneity of the native oxide interface and necessitates the use of non-ideal circuit elements (CPE) for accurate modeling.

The results demonstrate that in MOS structures with ultra-thin (native) oxides, the “insulating” layer acts as a field-dependent tunneling resistor. Impedance spectroscopy proves to be an effective method for distinguishing between capacitive accumulation and tunneling leakage regimes in such highly doped semiconductor systems.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The work was carried out within the framework of the projects supported by the Ukrainian Ministry of Education and Science (project No. 0125U000544).

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [D.S., Y.S., A.L.]; methodology, [D.S., M.K.]; investigation, [D.S., M.K., R.L.]; writing – original draft preparation, [D.S., R.L.]; writing – review and editing, [D.S., M.K., A.L.]; visualization, [D.S., M.K., R.L., Y.S., A.L.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] King, A. J., Weber, A. Z., Bell, A. T. (2023). Theory and Simulation of Metal–Insulator–Semiconductor (MIS) Photoelectrodes. *ACS Applied Materials & Interfaces*, *15*(19), 23024–23039. https://doi.org/10.1021/acsami.2c21114

[2] Helm, R., Egger, W., Corbel, C., Sperr, P., Butterling, M., Wagner, A., Liedke, M. O., Mitteneder, J., Mayerhofer, M., Lee, K. (2025). Defect Studies in Thin-Film SiO_2 of a Metal-Oxide-Silicon Capacitor Using Drift-Assisted Positron Annihilation Lifetime Spectroscopy. *Nanomaterials*, *15*(15), 1142. https://doi.org/10.3390/nano15151142

[3] Koch, J., Liborius, L., Kleinschmidt, P., Prost, W., Weimann, N., Hannappel, T. (2024). Impact of the Tip-to-Semiconductor Contact in the Electrical Characterization of Nanowires. *ACS Omega*, *9*(5), 5788–5797. https://doi.org/10.1021/acsomega.3c08729

[4] Macdonald, J. R., Johnson, W. B. (2018) *Fundamentals of Impedance Spectroscopy. Impedance Spectroscopy: Theory, Experiment, and Applications*, Third Edition, Hoboken, NJ: Wiley, 528 p. https://doi.org/10.1002/9781119381860.ch1

[5] Chemla, M., Dufrêche, J. F., Darolles, I., Rouelle, F., Devilliers, D., Petitdidier, S.,Lévy, D. (2005). Bias voltage dependent electrochemical impedance spectroscopy of p- and n-type silicon substrates. *Electrochimica Acta*, *51*(4), 665–676. https://doi.org/10.1016/j.electacta.2005.05.028

[6] Yeow, T., Sun, J., Yao, Z., Jaubert, J. -N., Musselman, K. P. (2019). Evaluation of impedance spectroscopy as a tool to characterize degradation mechanisms in silicon photovoltaics. *Solar Energy*, *184*, 52–58. https://doi.org/10.1016/j.solener.2019.03.088

[7] Xiao, H., Huang, S. (2010). Frequency and voltage dependency of interface states and series resistance in Al/SiO_2/p-Si MOS structure. *Materials Science in Semicondutor Processing*, *13*(5–6), 395–399. https://doi.org/10.1016/j.mssp.2011.05.009

[8] Kahraman, A., Yilmaz, E., Kaya, S., Aktag, A. (2015). Effects of post deposition annealing, interface states and series resistance on electrical characteristics of HfO_2 MOS capacitors. *Journal of Materials Science: Materials in Electronics*, *26*, 8277–8284. https://doi.org/10.1007/s10854-015-3492-3

[9] Pavlyk, B. V., Kushlyk, M. O., Slobodzyan, D. P. (2017). Origin of dislocation luminescence centers and their reorganization in p-type silicon crystal subjected to plastic deformation and high temperature annealing. *Nanoscale Research Letters*, *12*, 358. https://doi.org/10.1186/s11671-017-2133-6

[10] Glatthaar, R., Schmidt, F.-P., Hammud, A., Lunkenbein, T., Okker, T., Huster, F., Seren, S., Greven, B. C., Hahn, G., Terheiden B. (2023). Silver Metallization with Controlled Etch Stop Using SiO_x Layers in Passivating Contacts for Improved Silicon Solar Cell Performance. *Solar*, *7*(1), 2300491. https://doi.org/10.1002/solr.202300491

[11] Wiśniewski, P., Jasiński, J., Mazurak, A., Stonio, B., Majkusiak, B. (2021). Investigation of Electrical Properties of the Al/SiO_2/n^{++}-Si Resistive Switching Structures by Means of Static, Admittance, and Impedance Spectroscopy Measurements. *Materials*, *14*(20), 6042. https://doi.org/10.3390/ma14206042

[12] Pavlyk, B. V., Grypa, A. S., Lys, R. M., Slobodzyan, D. P., Didyk, R. I., Shykoryak, Y. A. (2009). Features of electrophysical characteristics of surface–barrier structures based on p-Si crystals [in Ukrainian]. *Physics and Chemistry of Solid State*, *10*(4), 935–938.

[13] Slobodzyan, D., Kushlyk, M., Lys, R., Matviishyn, I., Hryzohlazov, N., Shpotyuk, Ya., Luchechko, A. (2022). Radiative and magnetically stimulated change of the electrical properties of the SiO_2-Si structure with a high content of dislocation complexes. *Electronics and Information Technologies*, *20*, 74-86. https://doi.org/10.30970/eli.20.8

[14] Sze, S. M., Ng, K. K. (2006) *Physics of Semiconductor Devices*, 3rd ed.; Wiley-Interscience: Hoboken, NJ., 815 p. https://doi.org/10.1002/0470068329

[15] Lasia, A. (2002). Electrochemical impedance spectroscopy and its applications. In B.E. Conway, J.O. Bockris, & R.E. White (Eds.), *Modern aspects of electrochemistry*, *32*, 143–248. https://doi.org/10.1007/0-306-46916-2_2

ЕЛЕКТРИЧНИЙ ІМПЕДАНС СТРУКТУРИ МЕТАЛ-КРЕМНІЙ З ТОНКИМ ШАРОМ SiO_2

Дмитро Слободзян**, Маркіян Кушлик, Ярослав Шпотюк, Роман Лис, Андрій Лучечко
Кафедра сенсорної та напівпровідникової електроніки,
Львівський національний університет імені Івана Франка,
вул. Тарнавського, 107, 79017 Львів, Україна

АНОТАЦІЯ

Вступ. Структури метал-оксид-напівпровідник (МОН) на основі кремнію з діелектричним шаром SiO_2 широко використовуються в різноманітних пристроях сучасної електроніки. Тому дослідження електричних властивостей таких структур є актуальними. Особливу цінність при вивченні цих властивостей займає імпедансна спектроскопія, що дозволяє досліджувати механізми переносу заряду, властивості міжфазних границь та визначати параметри еквівалентних електричних схем заміщення.

Матеріали та методи. Досліджено зразки монокристалічного кремнію n-типу провідності з такими характеристиками: легуюча домішка – арсен, питомий опір – ρ = 0,003 Ом·см, товщина – 0,5 мм, площа – 30 мм^2. На поверхні такого зразка формувалася структура типу метал-напівпровідник-метал (Ag-SiO_2-Si-SiO_2-Ag).

Для дослідження частотних залежностей імпедансу сформованої структури було використано установку на основі RLC метра HIOKI IM3536.

Результати. Досліджено частотні залежності дійсної та уявної складових імпедансу в діапазоні частот від 4 Гц до 8 МГц структури Ag-SiO_2-Si-SiO_2-Ag за різних амплітуд збуджуючого електричного поля. Встановлено, що для всіх значень амплітуди сигналу збудження дійсна складова імпедансу *Re(Z)* зменшується зі збільшенням частоти та напруги зміщення. Частотні залежності уявної складової імпедансу *Im(Z)* демонструють наявність релаксаційних максимумів. Зі збільшенням амплітуди електричного поля величина цих максимумів значно зменшується. Спостережувані зміни імпедансу такої структури можуть бути пояснені на основі її багатошарової природи та властивостей високолегованого кремнію і тонких шарів природного оксиду кремнію.

Висновки. Дослідження структур Ag-SiO_2-Si-SiO_2-Ag з шарами природного оксиду за допомогою імпедансної спектроскопії виявило сильну нелінійну залежність електричних властивостей від амплітуди збуджуючого сигналу. Результати показують, що в МОН-структурах з надтонкими оксидними шарами, «ізоляційний» шар діє як полезалежний тунельний резистор. Імпедансна спектроскопія є ефективним методом для виявлення та розділення процесів ємнісного накопичення та тунельного витоку в таких сильно легованих напівпровідникових системах.

Ключові слова: кремній, імпедансна спектроскопія, МОН-структура, діаграма Найквіста.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
17 November, 2025	10 December, 2025	10 December, 2025	25 December, 2025

Electronics and Information Technologies, 2025, 32, 179–186
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC 004.94

COMPUTER MODELING OF ELECTROMAGNETIC WAVE SCATTERING BY NANOCOMPOSITES CONSIDERING THE AGGREGATION EFFECT

Igor Olenych**, Oleh Sinkevych, Yurii Olenych, Olena Aksimentyeva
Ivan Franko National University of Lviv
50 Dragomanov Str., 79005 Lviv, Ukraine

Olenych, I., Sinkevych, O., Olenych, Yu., & Aksimentyeva, O. (2025). Computer Modeling of Electromagnetic Wave Scattering by Nanocomposites Considering the Aggregation Effect. *Electronics and Information Technologies*, 32, 179–186. https://doi.org/10.30970/eli.32.13

ABSTRACT

Background. The increase in electromagnetic pollution requires the creation of new materials that can effectively shield radiation and mitigate its negative impact. Special attention is focused on composite materials due to their ability to absorb or scatter radiation. Rayleigh theory makes it possible to predict the intensity of scattered radiation by dispersed media in the case when the particle sizes are much smaller than the wavelength. Considering the possibility of nanoparticle aggregation due to stochastic processes can increase the prediction efficiency.

Model and Implementation Tools. The proposed nanocomposite model has the shape of a cube in which spherical particles of the same radius and volume are randomly arranged. The formation of agglomerates occurred when the distance between the centers of neighboring particles did not exceed their diameter. Additionally, the interaction of particles at a small distance between them, which causes their mutual attraction, was considered. Numerical experiments were implemented in Python using NumPy libraries for vectorized computations and cKDTree for efficient spatial neighbor search.

Results and Discussion. The average values of the number of filler particles of different volumes formed as a result of random aggregation were determined in the range of filler concentrations of 0.1–7% using computational experiments. It was found that the percentage of isolated initial nanoparticles decreases with increasing filler concentration. The largest number of formations contains only two nanoparticles. It was established that the scattering of electromagnetic waves in the IR and radio frequency ranges increases due to the aggregation of nanoparticles. The size and concentration of the initial nanoparticles have different effects on the change in the scattering cross-section. The mutual attraction and coupling of closely spaced particles additionally increase the Rayleigh scattering by dispersed media.

Conclusions. An averaged distribution of the formed agglomerates in a system of spherical nanoparticles by their volume was found as a result of a series of computational experiments. Based on Rayleigh theory, it has been demonstrated that the aggregation of nanoparticles causes an increase in the scattering of electromagnetic waves by dispersed media.

Keywords: Modeling, numerical experiment, dispersed medium, nanoparticle aggregation, Rayleigh scattering.

INTRODUCTION

The widespread application of wireless communication technologies and the ubiquitous use of electronic devices are not only significantly changing our society but also

© 2025 Igor Olenych et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

causing an increase in electromagnetic pollution and potential health hazards. Besides, increasing electromagnetic interference can degrade the performance of various electronic devices. Control over growing electromagnetic interference requires the creation of new materials that can effectively shield radiation and mitigate its negative impact, while taking into account other technological demands such as high strength-to-weight ratio, flexibility, corrosion resistance, etc. [1-3]. As a result, composite radiation-absorbing materials (RAMs) have become the focus of much research due to their ability to absorb or scatter electromagnetic waves, providing a variety of functionalities [4-6]. In particular, RAMs are used to prevent electromagnetic radiation leakage in medical equipment, to develop radiation-protective clothing, to ensure the compatibility of electronic devices with respect to electromagnetic interference, to improve the performance of communication systems, and to mask objects in the microwave and infrared ranges in military applications.

Typically, composite RAMs are produced by dispersing one or more types of absorbent fillers in a polymer matrix. Fillers have dielectric and magnetic losses, the frequency dependence of which is responsible for the effectiveness of the composite in absorbing and/or scattering electromagnetic waves. An additional advantage of composite RAMs is the close relationship between their structural and functional characteristics, which allows for tuning of electromagnetic interference shielding in the required frequency range by choosing the polymer matrix and filler type. The main attention of researchers is focused on radiation-absorbing fillers such as carbon black, graphene, carbon fibers and nanotubes, as well as materials with high dielectric permittivity (oxides, titanates, etc.) and magnetic permeability (ferrites and magnetodielectrics) [7-10]. To further improve the shielding characteristics of radiation-adsorbing composites, the influence of the structure and morphology of the filler on the properties of radiation attenuation is being extensively studied [11, 12]. Electromagnetic wave scattering by such composite materials also requires additional research.

Computational methods are usually used to elucidate the mechanisms of interaction of composite materials with radiation [13-15]. Most theoretical research has focused on determining the effective properties of disordered composites from first principles, which requires significant computational resources. Mie theory and Rayleigh theory for the case where particle sizes are much smaller than the wavelength make it possible to predict the intensity of scattered radiation by dispersed media. Considering the possibility of nanoparticle aggregation due to stochastic processes can increase the forecasting effectiveness. Therefore, the purpose of the work was to find the features of electromagnetic wave scattering by disordered composites with one type of filler based on computer simulation of a system of spherical nanoparticles in a host matrix.

MODEL DESIGN

A model of a nanocomposite volume element in the shape of a cube with edge length a, in which spherical particles of the same radius R_0 and volume V_0 are randomly arranged, is proposed in the paper. The dimensions of the cube are two to three orders of magnitude larger than the dimensions of the filler nanoparticles. For clarity, the model investigated in the work corresponds to a nanocomposite film with a thickness of 5 μm and a radius of filler nanoparticles in the range of 5–100 nm, which is consistent with the case of scattering of radiation in the IR or radio frequency ranges by a thin composite coating. The model can be applied when the wavelength is at least an order of magnitude larger than the particle size. The number of nanoparticles N_0 with radius R_0 in the considered element is determined by the volume concentration of the filler C_V as

$$N_0 = \frac{C_V a^3}{V_0}. \quad (1)$$

The position of each nanoparticle in the volume element is set randomly. Their aggregation into larger agglomerates was carried out similarly to the formation of percolation clusters in the nanotube system [16]. In the case when the distance d between the centers of neighboring nanoparticles does not exceed $2R_0$, the model assumes their aggregation into a larger structure, as shown in **Fig. 1**. The nanoparticles were moved apart by a distance of double the radius for the case $d < 2R_0$ to avoid overlapping of the initial nanoparticles and a decrease in the volume of the formed agglomerate. As a result, the volume of the agglomerate formed by the aggregation of k spherical components was kV_0.

The parameter δ was additionally introduced to take into account the interatomic and/or intermolecular interaction of nanoparticles (van der Waals forces). In this case, the nanoparticles were attracted to each other under the condition $d \le 2R_0+\delta$. The parameter δ value was in the range from 0.5 to 4 nm, which corresponds to the distances characteristic of van der Waals interactions.

A set of computational experiments was carried out for each value of concentration C_V in the 0.1–7% range and the radius R_0 of the nanoparticles in the range of 5–100 nm. As a result, the averaged distribution of the formed filler particles by their volume was found. The total number of initial nanoparticles in the volume element was in the 10^2–10^8 range and depended on their size and concentration. The effect of nanoparticle aggregation on the scattering of electromagnetic waves was investigated using the expression for Rayleigh scattering by dispersed media

$$\sigma_0 = \frac{24\pi^3 \nu V_0^2}{\lambda^4}\left(\frac{n_f^2 - n_m^2}{n_f^2 + 2n_m^2}\right)^2. \tag{2}$$

Here, σ_0 is the scattering cross-section, v is the number of particles with the volume V_0 in a unit volume, λ is the wavelength, n_f and n_m are the refractive indices of the filler material and the host matrix, respectively. Given that the total number of initial nanoparticles in the volume element for the chosen concentration C_V and radius R_0 does not change, the change in the Rayleigh scattering cross-section due to random aggregation of nanoparticles will be determined by the expression

$$\frac{\sigma}{\sigma_0} = \frac{1}{N_0}\sum_k N_k k^2, \tag{3}$$

where N_k is the number of particles formed as a result of the aggregation of k initial nanoparticles of volume V_0.

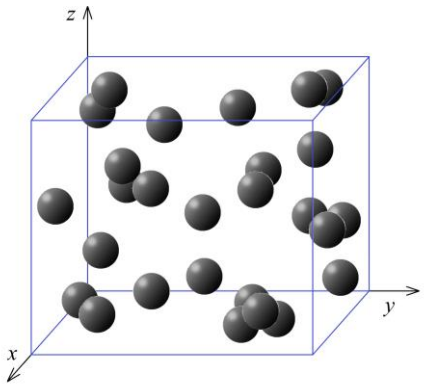


Fig. 1. Schematic representation of the composite material model taking into account nanoparticle aggregation.

IMPLEMENTATION TOOLS

The numerical experiments were implemented in Python using open-source scientific libraries such as NumPy for vectorized computations and SciPy's cKDTree for efficient spatial neighbor searches. Particle positions were generated with NumPy's random module and iteratively adjusted through custom overlap-resolution routines that operate on NumPy arrays. The cKDTree structure from SciPy was used to rapidly identify particle pairs within specified distance thresholds during each relaxation step. Cluster detection was performed through a Python implementation of the Union–Find algorithm, applied to neighbor pairs returned by cKDTree queries.

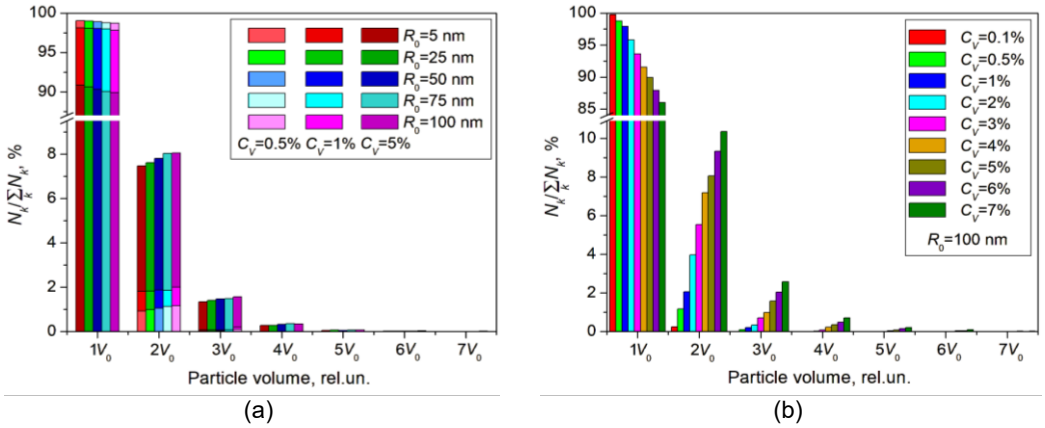
All simulations were executed over multiple concentration values and Monte-Carlo trials within a single Python workflow, producing cluster-size statistics for each scenario.

RESULTS AND DISCUSSION

The average values of the number of filler particles of different volumes formed because of random aggregation were determined through computational experiments. As can be seen in **Fig. 2**, the vast majority of the initial nanoparticles do not combine into larger aggregates and remain isolated for the investigated range of filler concentration. The fraction of isolated nanoparticles of volume V_0 is maximum (about 99.8%) for a concentration of 0.1% and decreases to 86.0% when the filler concentration increases to 7%. Conversely, the percentage of formed aggregates with a volume of $2V_0$ was increased from 0.2 to 10.3% with increasing filler concentration in the range of 0.1–7%. A similar trend was observed for particles with volumes of $3V_0$, $4V_0$, etc. Moreover, their share in the distribution decreased exponentially with increasing volume of formations. The percentage of particles with a volume exceeding $7V_0$ was less than 0.1% even for the maximum filler concentration within the study. It should be noted that the size of the initial nanoparticles had almost no effect on the distribution of the formed aggregates by their volume.

The found distribution of the number of formed filler particles by their volume made it possible to study the effect of random aggregation of the initial nanoparticles on the processes of Rayleigh scattering of electromagnetic waves by dispersed media using expression (3). **Fig. 3** shows the dependence of the ratio σ/σ_0 on the concentration C_V of the filler.

It has been established that the scattering of electromagnetic waves increases significantly due to the aggregation of nanoparticles. In particular, the ratio σ/σ_0 is exponentially increased with increasing nanoparticle concentration, which is associated with a greater number of larger formations. For a 7% concentration of nanoparticles in the host matrix, Rayleigh scattering increased by almost 1.5 times. A perceptible effect of the size of the



(a) (b)

Fig 2. Particle volume distribution for different values of the radius R_0 of the initial nanoparticles (a) and the filler concentration C_V (b).

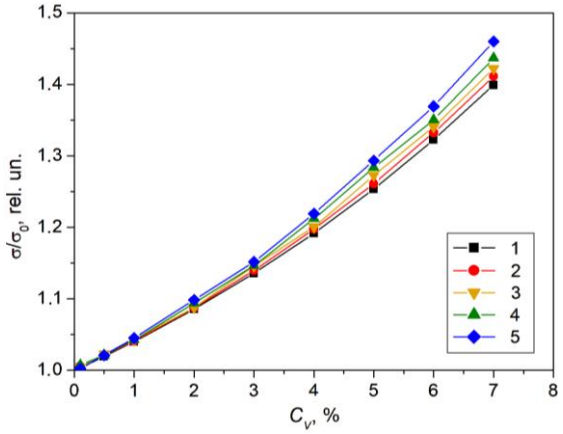


Fig 3. The dependence of the ratio σ/σ_0 on the filler concentration C_V for different values of the radius R_0 of the initial nanoparticles: 5 nm (1), 25 nm (2), 50 nm (3), 75 nm (4), and 100 nm (5).

initial nanoparticles on the aggregation-induced increase in the scattering of electromagnetic waves were observed only at a significant filler concentration. The dependencies of the ratio σ/σ_0 of the scattering cross-section on the radius R_0 of the initial nanoparticles for their various concentrations, shown in **Fig. 4**, confirm this trend. In the concentration range of 0.1–1%, no change in the value of the σ/σ_0 ratio was observed with increasing nanoparticle size, which is probably due to the small number of aggregated nanoparticles. Higher filler concentrations result in an almost linear increase in the intensity of scattered radiation with increasing R_0 compared to the case where all nanoparticles were isolated.

Consideration of the possibility of interaction of particles at a short distance between them ($2R_0 < d \le 2R_0+\delta$), which causes their mutual attraction and the formation of larger agglomerates, is an important aspect of studying the influence of particle aggregation on the scattering of electromagnetic waves by dispersed systems. **Fig. 5** shows the dependencies of the ratio σ/σ_0 of the scattering cross-sections on the parameter δ, which determines the limiting distance of interaction between particles.

An increase in Rayleigh scattering due to the coupling of closely spaced particles was found. In particular, an increase in the limiting distance of nanoparticle interaction leads to a greater number of agglomerates and, as a consequence, an increase in the ratio σ/σ_0. In addition, the obtained dependencies are more influenced by the concentration of nanoparticles than by their radius.

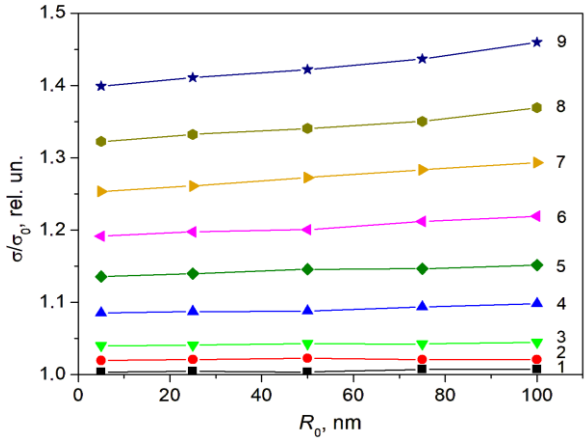


Fig. 4. Dependence of the ratio σ/σ_0 on the radius R_0 of the initial nanoparticles for different values of the filler concentration C_V: 0.1% (1), 0.5% (2), 1% (3), 2% (4), 3% (5), 4% (6), 5% (7), 6% (8), and 7% (9).

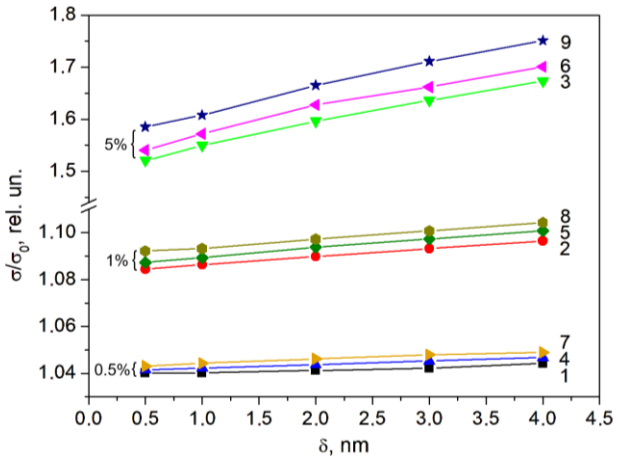


Fig 5. Dependence of the ratio σ/σ_0 on the limiting particle interaction distance δ for different values of the radius R_0 of the initial nanoparticles: 5 nm (1,2,3), 50 nm (4,5,6), and 100 nm (7,8,9). Curves (1,4,7), (2,5,8), and (3,6,9) related to C_V filler concentrations of 0.5, 1, and 5%, respectively.

CONCLUSION

In the work, the features of random aggregation of spherical nanoparticles in a dispersed medium were studied using computer simulation. As a result of computational experiments, an average distribution of the formed agglomerates by their volume was determined. It was established that the percentage of isolated initial nanoparticles decreases when the filler concentration increases. The formed agglomerates demonstrate the opposite trend. In addition, the share of the agglomerates in the distribution decreased according to an exponential law when their volume was increased. The radius of the initial nanoparticles has a weak effect on the distribution of the agglomerates by their volume.

Based on the numerical experiment, it was established that the scattering cross-section of electromagnetic waves in the IR and radio frequency ranges increases due to the aggregation of nanoparticles. Besides, the filler concentration has a greater effect on the aggregation-induced change in Rayleigh scattering than the radius of the initial nanoparticles. Conglutination of closely spaced particles additionally increases the radiation scattering by nanocomposites.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was partly supported by the Ministry of Education and Science of Ukraine [0124U001104].

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [IO, YO, OA]; methodology, [IO, OS, YO]; investigation, [IO, OS, YO]; writing – original draft preparation, [IO, YO]; writing – review and editing, [IO, OA]; visualization, [YO].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Cheng, J., Zhang, H., Ning, M., Raza, H., Zhang, D., Zheng, G., Zheng, Q., & Che, R. (2022). Emerging materials and designs for low- and multi-band electromagnetic wave

absorbers: the search for dielectric and magnetic synergy? *Adv. Funct. Mater.*, 32(23), 2200123. https://doi.org/10.1002/adfm.202200123

[2] Shu, J. C., Cao, M. S., Zhang, Y. L., & Cao, W. Q. (2023). Heterodimensional structure switching multispectral stealth and multimedia interaction devices. *Adv. Sci.*, 10(26), 2302361. https://doi.org/10.1002/advs.202302361

[3] Cao, M. S., Shu, J. C., Wang, X. X., Wang, X., Zhang, M., Yang, H. J., Fang, X. Y., & Yuanet, J. (2019). Electronic structure and electromagnetic properties for 2D electromagnetic functional materials in gigahertz frequency. *Ann. Phys*., 531(4), 1800390. https://doi.org/10.1002/andp.201800390

[4] Alegaonkar, A. P., Baskey, H. B., & Alegaonkar, P. S. (2022). Microwave scattering parameters of ferro–nanocarbon composites for tracking range countermeasures. *Mater. Adv*., 3, 1660-1672. https://doi.org/10.1039/D1MA00977J

[5] Jayalakshmi, C. G., Inamdar, A., Anand, A., & Kandasubramanian, B. (2019). Polymer matrix composites as broadband radar absorbing structures for stealth aircrafts. *J. Appl. Polym. Sci*., 136(14), 47241. https://doi.org/10.1002/app.47241

[6] Cheng J. Y., Zhang H. B., Xiong Y. F., Gao L. F., Wen B., Raza, H., Wang, H., Zheng, G., Zhang, D., & Zhang, H. (2021). Construction of multiple interfaces and dielectric/magnetic heterostructures in electromagnetic wave absorbers with enhanced absorption performance: a review. *J. Materiomics*, 7(6), 1233–1263. https://doi.org/10.1016/j.jmat.2021.02.017

[7] Folgueras, L. C., Alves, M. A., & Rezende, M. C. (2010). Microwave absorbing paints and sheets based on carbonyl iron and polyaniline: measurement and simulation of their properties. *Journal of Aerospace Technology and Management*, 2, 63-70. https://doi.org/10.5028/jatm.2010.02016370

[8] Yadav A., Tripathi, K. C., Baskey H. B., & Alegaonkar P. S. (2022). Microwave scattering behaviour of carbon black/ molybdenum di sulphide /cobalt composite for electromagnetic interference shielding application. *Materials Chemistry and Physics*, 279, 125766. https://doi.org/10.1016/j.matchemphys.2022.125766

[9] Joshi, A., Bajaj, A., Singh, R., Alegaonkar, P. S., Balasubramanian, K., & Datar, S. (2013). Graphene nanoribbon–PVA composite as EMI shielding material in the X band. *Nanotechnology*, 24, 455705. https://doi.org/10.1088/0957-4484/24/45/455705

[10] Wang, Y., Chen, Y. B., Wu, X. M., Zhang, W. Z., Luo, C. Y., & Li, J. (2018). Fabrication of MoS_2-graphene modified with Fe_3O_4 particles and its enhanced microwave absorption performance. *Adv. Powder Technol*. 29(3), 744–750. https://doi.org/10.1016/j.apt.2017.12.016

[11] Cheng, J., Jin, Y., Zhao, J., Jing, Q., Gu, B., Wei, J., Yi, S., Li, M., Nie, W., Qin, Q., Zhang, D., Zheng, G., & Che, R. (2024). From VIB- to VB-Group Transition Metal Disulfides: Structure Engineering Modulation for Superior Electromagnetic Wave Absorption. *Nano-Micro Lett*. 16, 29. https://doi.org/10.1007/s40820-023-01247-7

[12] Zhang, W., Zhang, X., Wu, H., Yan, H., & Qi, S. (2018). Impact of morphology and dielectric property on the microwave absorbing performance of MoS_2-based materials. *J. Alloys Compd*. 751, 34–42. https://doi.org/10.1016/j.jallcom.2018.04.111

[13] Torquato, S., & Kim, J. (2021). Nonlocal Effective Electromagnetic Wave Characteristics of Composite Media: Beyond the Quasistatic Regime. *Phys. Rev*. X, 11, 021002. https://doi.org/10.1103/PhysRevX.11.021002

[14] Kong, M., Cao, X. Y., Qi, Q., & Wu, X. L. (2025). An effective solution for analyzing the electromagnetic scattering characteristics of composite conducting dielectric objects under multiple angle incidence. *Sci. Rep*., 15, 26844. https://doi.org/10.1038/s41598-025-11786-1

[15] Sarkar T. K. & Jung B. H. (2004). Analysis of Scattering From Arbitrarily Shaped 3-D Conducting/Dielectric Composite Objects Using a Combined Field Integral Equation. *Journal of Electromagnetic Waves and Applications*, 18(6), 729-743. http://dx.doi.org/10.1163/156939304323105826

[16] Olenych, Y., Karbovnyk, I., & Klym, H. (2018). Computer simulation of field-controlled percolation in 3D system of straight nanotubes. *XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH),* pp. 48-51. https://doi.org/10.1109/MEMSTECH.2018.8365699

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ РОЗСІЮВАННЯ ЕЛЕКТРОМАГНІТНИХ ХВИЛЬ НАНОКОМПОЗИТАМИ З УРАХУВАННЯМ ЕФЕКТУ АГРЕГУВАННЯ

Ігор Оленич, **Олег Сінькевич**, **Юрій Оленич**, **Олена Аксіментьєва***
Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 м. Львів, Україна

АНОТАЦІЯ

Вступ. Збільшення електромагнітного забруднення потребує створення нових матеріалів, які можуть ефективно екранувати випромінювання та пом'якшувати його негативний вплив. Особлива увага дослідників зосереджена на композитних матеріалах, завдяки їх здатності поглинати або розсіювати електромагнітні хвилі. Теорія Релея дає змогу прогнозувати інтенсивність розсіяного дисперсними середовищами випромінювання у випадку, коли розміри частинок значно менші від довжини хвилі. Врахування можливості агрегування наночастинок внаслідок стохастичних процесів може підвищити ефективність прогнозування.

Модель і засоби реалізації. Запропонована модель нанокомпозиту має форму куба, в якому хаотично розташовані сферичні частинки однакового радіуса та об'єму. Утворення агломератів відбувалося коли відстань між центрами сусідніх частинок не перевищувала їхнього діаметра. Додатково було враховано взаємодію частинок на невеликій відстані між ними, яка зумовлює їх взаємне притягання. Числові експерименти були реалізовані на Python з використанням бібліотек NumPy для векторизованих обчислень та cKDTree для ефективного пошуку просторових сусідів.

Результати. Середні значення кількості частинок наповнювача різного об'єму, що утворилися в результаті випадкового агрегування, були визначені у діапазоні концентрацій наповнювача 0,1–7% за допомогою обчислювальних експериментів. Виявлено, що відсоток ізольованих вихідних наночастинок зменшується зі збільшенням концентрації наповнювача, а найбільша кількість утворень містить лише дві наночастинки. Встановлено, що розсіювання електромагнітних хвиль в ІЧ та радіочастотному діапазонах зростає за рахунок агрегування наночастинок. Розмір і концентрація вихідних наночастинок мають різний вплив на зміну перерізу розсіювання. Взаємне притягання і з'єднання близько розміщених частинок додатково підвищує розсіювання електромагнітних хвиль дисперсними середовищами.

Висновки. У результаті серії обчислювальних експериментів знайдено усереднений розподіл утворених агломератів у системі сферичних наночастинок за їх об'ємом. На основі теорії Релея продемонстровано, що агрегування наночастинок зумовлює збільшення розсіювання електромагнітних хвиль дисперсними середовищами.

Ключові слова: Моделювання, числовий експеримент, дисперсне середовище, агрегування наночастинок, розсіювання Релея.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
10 December, 2025	19 December, 2025	19 December, 2025	25 December, 2025

Збірник наукових праць

Електроніка та інформаційні технології

Electronics and information technologies

Випуск 32

2025

Підп. до друку 24.12.2025. Формат 70x100,16. Папір друк.
Друк на різогр. Гарнітура Times New Roman. Умовн. друк. арк. .
Тираж 100 прим. Зам. № .

Львівський національний університет імені Івана Франка.
79000 Львів, вул. Університетська, 1.

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців, виготівників і розповсюджувачів видавничої продукції. Серія ДК № 3059 від 13.12.2007 р.