UDC: 004.85:004.62 **DOI:** 10.31673/2412-9070.2026.058810

Yu. MELNYK[1], D.Sc., professor;
ORCID: 0000-0003-2307-1592
A. VOITKO[2], lecturer;
ORCID: 0009-0001-9521-0507
V. KUZMINYKH[2], PhD, associate professor;
ORCID: 0000-0002-8258-0816
V. KOLUMBET[2], PhD, senior lecturer,
ORCID: 0000-0002-0871-9402
[1]State University of Information and Communication Technologies, Kyiv
[2]National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»

ADAPTIVE DATA PROCESSING USING A MACHINE LEARNING MECHANISM

The relevance of this topic stems from the rapid growth in the number of heterogeneous and unstable data sources in modern information systems. Under such conditions, traditional approaches to organizing information retrieval no longer ensure sufficient efficiency, as they do not account for changes in the availability of sources, their response times, and the quality of the results obtained. The problem becomes particularly relevant in event-driven architectures, for which there are currently no universal mechanisms for adaptive control of the search process and selection of data sources. The increasing volume of publications requires innovative solutions that directly impact the quality of the results obtained and the overall efficiency of computational resource utilization. The aim of this article is to develop and implement reinforcement learning algorithms in event-driven systems to solve problems of adaptive evaluation and selection of data sources. To achieve this goal, an adaptive selection mechanism based on the Soft Actor-Critic (SAC) [9], [13] algorithm has been developed. An approach is proposed for forming a source state vector and a reward function based on behavioral and content metrics. This allows the system to dynamically adapt its search strategy depending on the characteristics of the query and the effectiveness of the sources in real time. Unlike static meta-search strategies, the developed system is capable of independently adapting to environmental changes during operation. A simulation environment was created, the results of which confirmed the unquestionable advantage of the adaptive approach: balance is achieved between result quality (increased relevance) and costs (reduced processing time and elimination of duplicates).

Keywords: adaptive source selection, event-driven systems, machine learning, reinforcement learning, Soft Actor-Critic, asynchronous data processing, ranking, deduplication, distributed search systems, reward function, state vector, action vector, scientific information aggregation, semantic search, relevance, dynamic control, query optimization, metasearch systems.

Introduction

Statement of the Problem. Modern information systems operate in an environment of constantly growing data volumes and an increasing number of external information sources. This is particularly evident in systems for searching and aggregating scientific information, which constantly interact with a specific set of external services and scientific data sources (scientific libraries, academic search engines, digital archives, specialized databases). Today, distributed information systems operate in an environment of heterogeneous data sources that differ in structure, response speed, and, of course, the relevance of their results.

Most existing systems use static mechanisms for working with data sources. Most often, either a single centralized index is queried, or all available sources are queried simultaneously. This approach has several significant drawbacks: increased query processing time, excessive system load, duplicate results, and inefficient use of computational resources. In addition, it should be noted that source

© Melnyk Yu., Voitko A., Kuzminykh V., Kolumbet V., 2026

characteristics can change dynamically: sources may temporarily operate more slowly, return incomplete results, or be less relevant for a specific type of query. This is precisely why modern distributed systems require adaptive decision-making mechanisms that automatically identify the most effective sources based on query type and the current environment state.

Analysis of recent research. An analysis of academic and commercial scientific information aggregators (Google Scholar, Semantic Scholar) [4] shows that they are based on their own centralized indexes. Queries are processed by a single search engine without an explicit selection of sources for each query, and the ranking of results depends on the number of citations and the completeness of the text. Since the model is proprietary, the user cannot access individual sources or influence their selection. Specialized databases (IEEE Xplore, PubMed, Scopus) have their own indexes and APIs; however, in corporate settings, it is necessary to combine searches across multiple such sources.

Meta-search engines (WorldWideScience, BASE) expand the scope of information resources by simultaneously searching multiple independent sources. However, they have limitations, including significant duplication of results and inefficient resource use, as queries are sent to all sources regardless of their current status. Federated search systems (Primo, Summon, EBSCO Discovery Service) do not use a centralized index, but their overall processing time depends largely on the slowest service.

Traditional approaches to information retrieval rely on statistical ranking methods, such as TF-IDF and the BM25 algorithm [6]. These have been further developed into neural approaches (embedding-based retrieval, DPR) and Retrieval-Augmented Generation (RAG) [1] systems. However, most of them are focused on retrieving documents from a single index or a fixed set of sources.

The development of intelligent approaches has led to the application of reinforcement learning algorithms. The Deep Q-Network (DQN) [8] algorithm is well-suited for discrete action spaces; however, for an adaptive selection system with multidimensional parameters (source priorities, ranking weights), its use requires discretization, which complicates training. The Proximal Policy Optimization (PPO) algorithm is an on-policy method that requires substantial data and many iterations. Therefore, the most promising algorithm is Soft Actor-Critic (SAC) [9], an off-policy method that supports a continuous action space and uses entropy regularization to balance exploration of new options with exploitation of effective strategies.

A summary of the comparative analysis of existing approaches is presented in table 1.

Table 1

Comparative analysis of existing approaches

Criterion	Centralized scientific systems (Google Scholar, Semantic Scholar)	Meta-search engines (WorldWideScience, BASE)	RAG-approaches (LangChain + Retriever)
Architecture	Centralized Index	Aggregation of external sources	Retriever + LLM
Selecting sources at runtime	Not in use	Call all sources	Fixed retriever
Forming a subset of sources	Not available	Not available	Limited
Consideration of behavioral characteristics of sources	Not available	Not available	Not available
Optimizing the number of calls	Internal	Not available	Partial
On-the-job training	Not in use	Not in use	Partially possible

Purpose of the article. To develop a model and algorithm for consolidating information from a set of large-scale, heterogeneous sources, as well as to integrate a reinforcement learning algorithm (Soft Actor-Critic) into event-driven systems for the adaptive evaluation and selection of data sources [20]. This will allow for real-time maximization of the relevance of the retrieved documents to the query with a minimum number of source accesses.

Main section

Event-Driven Architecture in Distributed Systems. To build highly scalable information systems, it is advisable to use Event-Driven Architecture (EDA) [2]. A key feature of this approach is asynchronous interaction between components via message brokers and events [19]. This ensures loose coupling of modules, significantly improves the system's fault tolerance, and simplifies its horizontal scaling.

The architecture of the developed system consists of several logically separate layers: the client-side (Vue 3, TypeScript, Vite), the gateway service (Python, FastAPI) [10], the processor service (the system core with RL algorithms), a set of external data sources, and the infrastructure layer (SQLite, SQLAlchemy). Interaction between all components is implemented via the Apache Kafka [12] message broker.

The detailed architecture is shown in fig. 1.

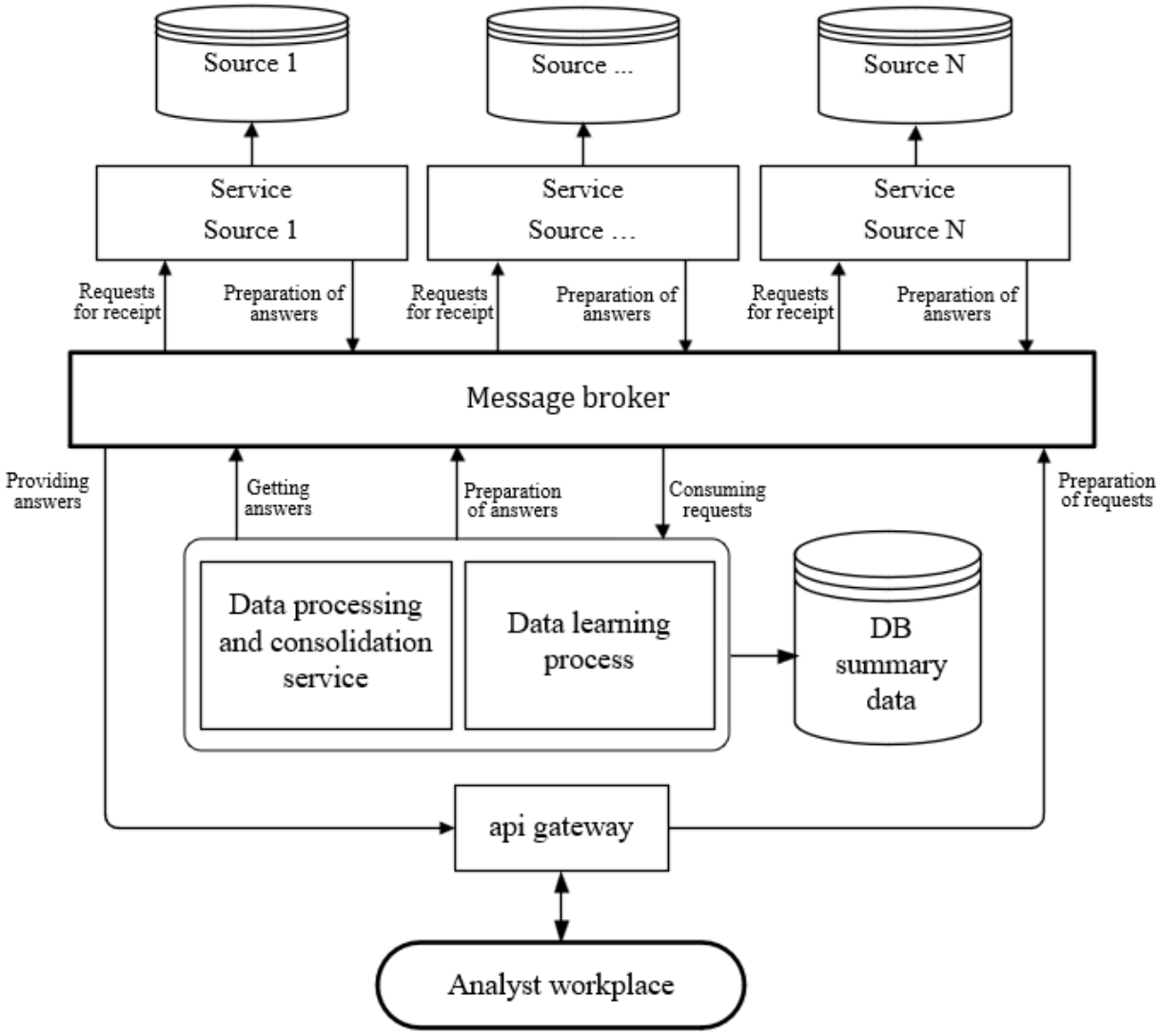


Fig. 1. Software Architecture

Request processing is organized as a sequential flow:

1. The user sends an HTTP POST request to the gateway component, which performs initial data validation.

2. The request is published to the Kafka topic `requests`.

3. The processor service asynchronously consumes messages and performs a full processing cycle: querying sources via a system of adapters (integrated real CrossRef [15] and arXiv [16] APIs, as well as a set of simulated sources), data normalization, deduplication, calculation of relevance scores, and adaptive source selection using a machine learning agent.

4. The results are published to the responses topic, from which the gateway retrieves the formated response and returns it to the client.

An important architectural feature is the use of the adapter pattern. All sources operate through a single abstract interface, which allows the processor service to remain independent of any specific protocol or response format used by individual data providers. The project structure includes the gateway, processor, sources (implementing artificial sources with their own delay profiles and success probabilities), and tests (over 80 standalone unit tests using pytest) directories.

Reinforcement learning model: Soft Actor-Critic (SAC). Soft Actor-Critic is used as the main adaptive selection algorithm. This is an off-policy algorithm based on the principle of maximum entropy, optimized for tasks with continuous action spaces [2]. The algorithm's operation is based on the agent's constant interaction with the environment through state and action vectors [17].

State vector. It is generated dynamically and reflects the system's current context. All features are normalized to the interval $[0,1]$ The vector contains 30 features, divided into:

1. *Query features:* presence of an author filter (0/1), presence of a year filter (0/1), normalized number of keywords, encoded topic tag (topic_tag).

2. *Source features (for each source N):* normalized response latency (latency), percentage of successful calls (success_rate), unique items contributed (unique items contributed), time since the last call (recency/time since last selected).

3. *Global features:* duplicate ratio in the system (duplicate_ratio), average processing time (processing_time), and previous reward estimate.

Action vector. Generated by the Actor component (a multilayer neural network with two hidden layers of 64 neurons each) and has a dimension of $N + 5$ components:

1. *Source priorities:* $p_1, p_2, \dots, p_N$ - determine the priority of each available source.

2. *Call budge*t: a continuous value converted into an integer representing the number of active sources that need to be queried.

3. *Ranking weights:* $w_{keyword}, w_{author}, w_{recency}, w_{latency}$ - dynamically determine the signifycance of the corresponding criteria during the final ranking of the retrieved documents.

Reward Function and Training. After an action is executed, the system calculates the actual results: relevance, duplicate count, and latency. These metrics are converted into a numerical reward. Next, the experience tuple (state, action, reward, next_state) is added to the Replay Buffer, which is limited to 10,000 transitions. Training is performed in batches (batch_size = 32).

The formation of the target expected reward is based on the use of two critics (Twin Critics) to avoid overfitting the Q-function:

$$y = r + \gamma \times (\min(Q_1(s', a'), Q_2(s', a')) - \alpha \log \pi(a'|s')) , \quad (1)$$

where r is the received reward; γ is the discount factor; Q_1, Q_1 are independent estimates from two critic networks for the next state s' and action a'; $\alpha \log \pi(a'|s')$ is the entropy term, which rewards greater stochasticity in the policy.

SAC updates Critic 1 and Critic 2 by training them to predict the target value y using the mean squared error:

$$L_{Q_i} = MSE(Q_i(s, a) - y)^2 \quad (2)$$

Target networks for critics are updated gradually using a small r value (for example, 0.005):

$$\theta_{target} \leftarrow \tau \times \theta + (1 - \tau) \times \theta_{target} \quad (3)$$

This approach encourages the agent not only to use already known effective strategies but also to explore new options, which is critically important in an environment where source characteristics are constantly changing.

The SAC agent's life cycle. To avoid unstable behavior at the start, the system implements a controlled training life cycle with three modes (table 2).

Table 2

System Operating Modes

Mode	Condition based on experience (buffer_size)	Description
Baseline	< 200 transitions	It queries all sources; SAC does not influence the selection of sources
Warmup	200 – 999 transitions	SAC is already learning and influencing the rankings, but all sources are still being called
Adaptive	>= 1000 transitions	SAC fully controls the selection of a subset of sources and the determination of ranking weights

During the Baseline and Warmup phases, the algorithm accumulates statistics in the Replay Buffer (gaining valuable experience without the risk of losing potentially useful documents) [8]. After transitioning to Adaptive mode, the agent takes full control over selecting the optimal number of queries and ranking weights, optimizing time allocation while maintaining response relevance [21]. Additionally, a controlled exploration mechanism is applied: with a certain probability, a random source is added to the subset of sources selected by the agent, enabling the agent to update its knowledge of its current state.

Simulation results and performance evaluation. To evaluate the effectiveness of the proposed approach, comparative modeling of three system operating modes was performed under identical experimental conditions. The modeling was conducted on a set of 500 synthetic queries (across various subject categories, random keyword combinations, and author/year filters). All modes received an identical set of queries.

Evaluation criteria: relevance of results, query processing time (performance), average number of retrieved sources (resource cost), and proportion of duplicates (redundancy). The simulation results are presented in table 3.

Table 3

Results of the system simulation

Method	Relevance	Processing time, ms	Average number of sources called	Duplicates
Baseline (all sources)	0.5945	1458.3	8.00	0.259
Random (stochastic approach)	0.6115	1152.7	4.67	0.240
Adaptive (SAC) (adaptive selection)	0.6218	1104.3	4.24	0.235

The results demonstrate a consistent and convincing advantage of the adaptive mode over the baseline approach, which involves fully querying all sources. Using a trained SAC agent reduced the computational cost of a query (on average, 4.24 sources were retrieved instead of 8.00), resulting in a significant reduction in latency [11]. At the same time, the relevance of the results not only did not decrease but, on the contrary, increased to 0.6218, since intelligent selection allowed for the rejection of sources that generated irrelevant "noise" or duplicates (the proportion of duplicates decreased to 0.235).

A comparison with the stochastic approach (Random), which randomly reduces the number of calls, demonstrates that the improvement in performance in adaptive mode is not due to the mere mechanical reduction of the source budget, but rather to the controllability and intelligence of the selection based on the current system state, source metrics, and the characteristics of the specific query.

The client-side part of the application (Vue.js) provided tools not only for performing searches but also for monitoring the agent's operation (SAC Runtime, SAC Evaluation). The system displays the reasons behind decisions to the user: which sources were queried, which weights were applied to the ranking, the latency level, and the execution status for each API. This makes the ML model's behavior transparent (explainable AI) and enables the system to be used in practice by corporations.

Conclusions

As part of this work, software was developed and tested for adaptive evaluation and selection of data sources in event-driven systems using reinforcement learning. The implementation of an event-driven architecture (via the Apache Kafka broker) [12] enabled efficient asynchronous communication, loose coupling among components, and high horizontal scalability.

A mechanism for adaptive source selection based on the Soft Actor-Critic (SAC) algorithm has been developed. A comprehensive mathematical model was constructed, including a state vector (source behavioral metrics and query content characteristics), a multidimensional action vector (priorities, call budget, ranking weights), and a dynamic reward function.

The results of experimental modeling confirm that, compared to the baseline approach (static polling of all available sources), the use of the adaptive method ensures:

- An increase in the overall relevance of results by ≈4.6%.
- A reduction in query processing time (latency) by ≈24%.
- A reduction in the number of calls to external sources by ≈47%, which optimizes resource consumption and relieves excess load on the API.
- A reduction in the number of duplicates by ≈9%, which improves the quality of the final document set.

Compared to the stochastic approach, a ≈1.7% increase in relevance and an additional ≈4% reduction in processing time were also achieved. The results obtained convincingly confirm the feasibility of using adaptive ML approaches in distributed information retrieval systems, R&D departments, and research institutions that operate with a significant number of unstable and heterogeneous data sources.

Author contributions

Yurii MELNYK - problem statement and formulation of the research aim, review of existing systems for scientific information retrieval and aggregation, justification of the choice of the reinforcement learning algorithm, scientific supervision, formulation of conclusions; Andrii VOITKO – development of the client side of the software and tools for displaying the agent's operation, connection of the CrossRef and arXiv data sources, software testing; Valerii KUZMINYKH - development of the event-driven system architecture using the Apache Kafka message broker, organization of data collection from external sources, analysis of the obtained results, editing of the manuscript; Vadym KOLUMBET - development of the mathematical model for adaptive data source selection, definition of the state vector, action vector and reward function, formalization of the SAC agent training process, software implementation of the algorithm, simulation modelling and processing of its results, preparation of the manuscript.

Declaration on artificial intelligence

The authors confirm that artificial intelligence technologies were used solely as the object of research and as a tool for experimental analysis, and that no generative models were used to create the text of the manuscript.

Conflict of interest

The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship-related or any other conflict that could have influenced the research and its results presented in this article.

References

1. Lewis P., Perez E., Piktus A. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems (NeurIPS 2020). — 2020. — Vol. 33. — P. 9459–9474.

2. Kleppmann M. Designing Data-Intensive Applications. — Sebastopol: O'Reilly Media, 2017. — 616 p.

3. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. — 2nd ed. — Cambridge: MIT Press, 2018. — 552 p.

4. Semantic Scholar API Documentation. URL: https://www.semanticscholar.org/product/api/ (дата звернення: 03.05.2026).

5. Muse® Federated Search. URL: https://www.edulib.com/products/the-muse-platform/muse-federated-search/ (дата звернення: 04.05.2026).

6. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond // Foundations and Trends in Information Retrieval. — 2009. P. 333–389.

7. Ban Y., He J., Jiang B., Yang C. Neural Contextual Bandits for Personalized Recommendation // arXiv preprint arXiv:2312.14037. — 2023.

8. Mnih V., Kavukcuoglu K., Silver D. et al. Human-level Control through Deep Reinforcement Learning // Nature. — 2015. — Vol. 518. — P. 529–533.

9. Haarnoja T., Zhou A., Abbeel P., Levine S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor // Proceedings of the 35th International Conference on Machine Learning (ICML). — 2018. — P. 1861–1870.

10. Python Documentation. URL: https://docs.python.org/uk/3/tutorial (дата звернення: 01.05.2026).

11. PyTorch Documentation. URL: https://pytorch.org/docs/ (дата звернення: 04.05.2026).

12. Apache Kafka Documentation. URL: https://kafka.apache.org/documentation/

13. Haarnoja T., Zhou A., Abbeel P., Levine S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor // Proceedings of the 35th International Conference on Machine Learning (ICML), 2018.

14. Soft Actor-Critic Reinforcement Learning algorithm. URL: https://www.geeksforgeeks.org/deep-learning/soft-actor-critic-reinforcement-learning-algorithm/

15. Crossref REST API Documentation. URL: https://api.crossref.org/ (дата звернення: 07.05.2026).

16. arXiv API User Manual. URL: https://info.arxiv.org/help/api/

17. Kuzminykh V.O., Otrokh S.I., Xu B., Taranenko R.A., Kublii L.I. Eventoriented architecture in the system for processing large data streams. Зв'язок, No 1, 2023, pp.22-27, ISSN 2412-9070, DOI: 10.31673/2412-9070.2023.012126

18. Kuzminykh V., Xu B. The influence of current results in an event-oriented data collection system ЗВ'ЯЗОК, No 3 (169), 2024 pp.18-22, DOI: 10.31673/2412-9070.2024.031822 https://con.dut.edu.ua/index.php/communication/article/view/2774/2667

19. Danylchenko V., Voitko A., Kuzminykh V., Kolumbet V. Modern Challenges of Disinformation in Media: Anomaly Detection in Social Network Metrics Using Machine Learning Models. Journal "Zvyazok", Issue No.2 (174), 2025, pp. 87-93. DOI:10.31673/2412-9070.2025.027701. https://con.dut.edu.ua/index.php/communication/article/view/2860/2755

20. Koval, O.V., Kuzminykh, V.O., Husyeva, I.I., Xu, B., Zhu, S. Construction Optimizing of Search and Data Processing Scerario Based on a Graphical Model, CEUR Workshop Proceedings, 2023, 3887, pp. 63–72 Activity Level Assessment CEUR Workshop Proceedings, № 3503,2022, pp. 52–61 ISSN:1613-00731.https://ceur-ws.org/Vol-3503/paper6.pdf.
https://www.scopus.com/record/display.uri?eid=2-s2.0-85215525225&origin=recordpage

21. Valeriy Kuzminykh, Oleksandr Koval, Yevhen Havrylko, Beybey Xu, Shiwei Zhu... Synchronization of event-driven management during data collection. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska.Vol. 14 No. 4 (2024) 2024, 14(4), p. 121–129, ISSN: 2083-0157/https://www.scopus.com/authid/detail.uri?authorId=57201372342
http://doi.org/10.35784/iapgos.6656

Ю. В. Мельник, А. С. Войтко, В. А. Кузьмінних, В. П. Колумбет

АДАПТИВНА ОБРОБКА ДАНИХ З ВИКОРИСТАННЯМ МЕХАНІЗМУ МАШИННОГО НАВЧАННЯ

Актуальність теми зумовлена стрімким зростанням кількості неоднорідних і нестабільних джерел даних у сучасних інформаційних системах. У таких умовах традиційні підходи до організації пошуку інформації вже не забезпечують достатньої ефективності, оскільки не враховують змін у доступності джерел, швидкості їх роботи та якості отриманих результатів. Особливої актуальності проблема набуває у подійно-орієнтованих архітектурах, для яких наразі відсутні універсальні механізми адаптивного керування процесом пошуку та вибору джерел даних. Збільшення обсягів публікацій вимагає новітніх рішень, що безпосередньо впливають на якість отриманих результатів та загальну ефективність використання обчислювальних ресурсів. Метою статті є розробка та впровадження алгоритмів машинного навчання з підкріпленням в подійно-орієнтовані системи для вирішення задач адаптивної оцінки та вибору джерел даних. Для досягнення цієї мети розроблено механізм адаптивного вибору на основі алгоритму Soft Actor-Critic (SAC) [9], [13]. Запропоновано підхід до формування вектора стану джерел та функції нагороди на основі поведінкових і змістовних метрик. Це дозволяє системі динамічно адаптувати стратегію пошуку залежно від характеристик запиту та ефективності джерел у режимі реального часу. На відміну від статичних стратегій метапошуку, розроблена система здатна самостійно адаптуватися до змін середовища у процесі роботи. Створено середовище моделювання, результати якого підтвердили беззаперечну перевагу адаптивного підходу: забезпечено баланс між якістю результатів (підвищення релевантності) та витратами (скорочення часу обробки та усунення дублікатів).

Ключові слова: адаптивний вибір джерел, подійно-орієнтовані системи, машинне навчання, навчання з підкріпленням, Soft Actor-Critic, асинхронна обробка даних, ранжування, дедуплікація, розподілені системи пошуку, функція нагороди, вектор стану, вектор дії, агрегація наукової інформації, семантичний пошук, релевантність, динамічне керування, оптимізація запитів, метапошукові системи.

Надійшла до редакції: 15.07.2026
Прийнята до друку: 05.08.2026
Опубліковано: 30.10.2026
© 2026 Melnyk Yu., Voitko A., Kuzminykh V., Kolumbet V.
Цей матеріал ліцензовано за умовами CC BY 4.0.https://creativecommons.org/licenses/by/4.0/