https://doi.org/10.31891/2307-5732-2025-351-17
УДК 004.8:004.4

ЛУЧКЕВИЧ МИХАЙЛО
Національний університет «Львівська політехніка»
https://orcid.org/0000-0002-2196-252X
e-mail: luchkevychmm@gmail.com
ШАКЛЕІНА ІРИНА
Національний університет «Львівська політехніка»
https://orcid.org/0000-0003-0809-1480
e-mail: ioshakleina@gmail.com
ДУДА ОЛЕКСІЙ
Тернопільський національний технічний університет імені Івана Пулюя
https://orcid.org/0000-0003-2007-1271
e-mail: oleksij.duda@gmail.com

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ DEVOPS ЗА РАХУНОК ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ТА МАШИННОГО НАВЧАННЯ

У статті досліджено підвищення ефективності DevOps-процесів шляхом інтеграції технологій штучного інтелекту та машинного навчання. Розглянуто основні виклики, з якими стикаються традиційні DevOps-підходи, включаючи обмеженість моніторингу, складність управління CI/CD-процесами, значний обсяг логів та інцидентів, а також неефективне масштабування ресурсів у хмарних середовищах. Акцентовано увагу на важливості впровадження інтелектуальних методів аналізу логів, прогнозування збоїв, автоматизації розгортання програмного забезпечення та оптимізації балансування навантаження.

Особливу увагу приділено використанню алгоритмів глибинного навчання, обробки природної мови та прогнозування часових рядів для покращення виявлення аномалій і забезпечення більш точного прогнозування навантаження. Окремо розглянуто механізми оптимізації процесів CI/CD за допомогою reinforcement learning, що дозволяє автоматично адаптувати налаштування пайплайнів відповідно до змінних умов роботи. Використання методів кластеризації та розпізнавання патернів у коді дозволяє виявляти потенційні помилки ще до етапу тестування, що суттєво знижує ризики збою при розгортанні нових версій. Впровадження інтелектуальних систем моніторингу дає можливість оперативно виявляти відхилення від нормальної роботи, що сприяє проактивному усуненню проблем і мінімізує людське втручання у критичні моменти.

У дослідженні також розглянуто перспективи подальшого розвитку, включаючи розробку інтегрованих платформ, що поєднують різні ШІ-моделі для моніторингу, управління ресурсами та автоматизації розгортання. Додатково проаналізовано застосування генеративних моделей ШІ для автоматизації кодування, зокрема створення оптимізованих тестових сценаріїв і автоматичного виправлення помилок у коді. Окрему увагу приділено можливості створення автономних DevOps-агентів, які можуть самостійно управляти життєвим циклом програмного забезпечення, виконувати аналіз ефективності розгортань та приймати рішення про відкат змін або подальшу оптимізацію ресурсів.

Ключові слова: DevOps, штучний інтелект, машинне навчання, автоматизація, моніторинг, управління ресурсами.

MYKHAILO LUCHKEVYCH
IRYNA SHAKLEINA
Lviv Polytechnic National University
OLEKSII DUDA
Ternopil Ivan Puluj National Technical University

INCREASING THE EFFICIENCY OF DEVOPS THROUGH THE USE OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

The article investigates the efficiency of DevOps processes by integrating artificial intelligence and machine learning technologies. Traditional DevOps approaches face limited monitoring, the complexity of managing CI/CD processes, a significant amount of logs and incidents, and inefficient scaling of resources in cloud environments. Attention is focused on implementing intelligent methods for analyzing logs, predicting failures, automating software deployment, and optimising load balancing.

Particular attention is paid to deep learning, natural language processing, and time series forecasting algorithms to improve anomaly detection and provide more accurate load forecasting. Mechanisms for optimising CI/CD processes using reinforcement learning, which allows for automatically adapting pipeline settings to changing operating conditions, are also discussed. Using clustering and pattern recognition methods in the code allows the detection of potential errors even before the testing stage, significantly reducing the risk of failure when deploying new versions. The introduction of intelligent monitoring systems allows for the prompt detection of deviations from normal operation, which facilitates proactive problem solving and minimises human intervention at critical moments.

The study also considers prospects for further development, including developing integrated platforms that combine different AI models for monitoring, resource management, and deployment automation. In addition, the article analyses the generative AI models application for coding automation, including the creation of optimized test scenarios and automatic correction of errors in the code. Particular attention is paid to the creating possibility autonomous DevOps agents that can independently manage the software life cycle, analyse the effectiveness of deployments, and make decisions on rolling back changes or further optimising resources.

Keywords: DevOps, artificial intelligence, machine learning, automation, monitoring, resource management.

Стаття надійшла до редакції / Received 11.04.2025
Прийнята до друку / Accepted 26.04.2025

Problem statement

Modern software development methodologies require a high level of automation, rapid integration of changes, and effective process management. DevOps is an approach that aims to optimise the software development life cycle by automating and integrating the development and operations processes. However, with the growing complexity of IT infrastructure, the software deployments number is increasing significantly, which poses significant challenges to traditional DevOps practices. For example, in large organisations, releases can occur dozens of times daily, making monitoring, testing, and managing potential incidents difficult. In addition, ensuring the continuous operation of services requires high accuracy in predicting possible failures, automating troubleshooting, and optimising resource utilisation. Traditional DevOps approaches face several challenges that make managing the software lifecycle difficult. One of the main problems is the large volume of logs and incidents. DevOps systems generate data vast amounts, including logs, metrics, and security events, which makes it difficult to analyse them manually. Identifying anomalies, determining the causes of failures, and finding correlations in this information flow become time-consuming tasks. Another critical aspect is the low predictability of failures and incidents. Despite modern monitoring systems, most failures are detected only after they occur, which can lead to extended downtime and significant financial losses.

Optimising CI/CD processes also remains a challenge. Traditional approaches to configuring pipelines are based on static rules that do not consider the variability of operating conditions. This leads to inefficient deployment processes, increasing time and resource costs. An equally important challenge is inefficient resource management. Dynamic load scaling and balancing are often performed manually or according to predefined parameters, which does not allow for a quick response to changes in load. As a result, resources may be overused or, conversely, service performance may decline. In this regard, there is a need to implement intelligent systems that can analyse large amounts of data, automate decision-making, and increase the DevOps efficiency. Artificial intelligence and machine learning technologies can be crucial factors in solving these problems.

Analysis of research and publications

Recent studies [1-5] in the field of DevOps show that the introduction of artificial intelligence (AI) and machine learning (ML) technologies can significantly improve the efficiency of DevOps processes. The development of automated systems for detecting anomalies, predicting possible failures, optimising CI/CD pipelines, and managing resources opens up new opportunities to improve the stability and performance of IT infrastructure [6]. Research confirms that implementing machine learning algorithms in DevOps helps improve process efficiency. Automatic log analysis and identifying root causes of failures significantly reduce the average recovery time [7]. In addition, adaptive testing and early detection of errors at the integration stage allow for optimising deployment processes, which minimises the risk of failures in the production environment [8].

A study [9] indicates that integrating AI into DevOps enhances resource allocation efficiency, maintains application stability, and minimizes downtime. In particular, AI models can automatically scale the infrastructure according to real-time needs and fix errors independently. Article [10] discusses the challenges of integrating machine learning models into production environments. Particular attention is paid to the issues of monitoring model performance, managing model deviations, and the need to retrain models. The article emphasises the importance of implementing scalable DevOps pipelines to ensure continuous monitoring and effective management of ML model performance.

Article [11] discusses the problems of scaling DevOps practices to support distributed machine learning in large organisations. The study focuses on challenges such as data management, model training efficiency, infrastructure orchestration, and team collaboration. The authors propose solutions, including containerisation, orchestration tools, automated testing, and CI/CD pipelines to optimise large-scale ML operations. Article [12] examines how DevOps has evolved into MLOps, with a particular focus on using MLOps for predicting electricity prices a day in advance. The authors explore how DevOps practices can be tailored for ML systems, enabling seamless integration and continuous delivery in machine learning workflows.

Another important aspect is the use of NLP-based computational techniques to automate incident analysis [13]. The research demonstrates that incorporating NLP models into DevOps processes enables quicker categorization and handling of technical support requests, thereby minimizing response time to critical incidents. Modern developments also include generative AI models to automate code writing and test creation [14, 15].

Thus, modern research confirms that using artificial intelligence technologies in DevOps can significantly improve the efficiency of development and operation processes. However, further research and experimentation are needed to achieve maximum benefit.

Formulating the objectives of the article

The aim of the work is to: explore the potential of artificial intelligence and machine learning technologies to enhance the effectiveness of DevOps processes. The study seeks to evaluate the practicality and impact of these technologies in real DevOps settings and identify methods for incorporating AI solutions into contemporary software development and operational management tools.

Summary of the main material

Studies [16-18] show that traditional DevOps approaches have some limitations, including reactive monitoring, manual configuration of CI/CD processes, and static resource management methods. Simultaneously, the application of artificial intelligence and machine learning can greatly enhance DevOps efficiency by automating log analysis, streamlining pipelines, and proactively addressing incidents. To better understand the benefits of DevOps process intelligence, Table 1 compares traditional DevOps and DevOps with AI.

Table 1

Comparison of traditional DevOps and DevOps with AI

Parameter	Traditional DevOps	DevOps with the use of AI
Monitoring	Reactive, analysis after incidents	Proactive, failure prediction
CI/CD	Fixed rules, no adaptation	Automatic optimisation of pipelines
Resource management	Static settings, manual scaling	Dynamic management based on forecasts
Analysing logs	Manually or by fixed rules	Automatic anomaly detection

Automation of monitoring and anomaly detection. One of the key areas of AI application in DevOps is automated monitoring. By applying machine learning models to analyze logs, it becomes possible to identify unusual events and potential failures ahead of time. Traditional approaches to monitoring are often based on setting strict thresholds and fixed rules, which can lead to many false positives or, conversely, miss essential incidents. Machine learning enables the development of adaptive monitoring systems [19] that can learn from past data and adapt to evolving operational conditions. Various machine learning algorithms are applied depending on the type of data and the monitoring objectives (Fig. 1).

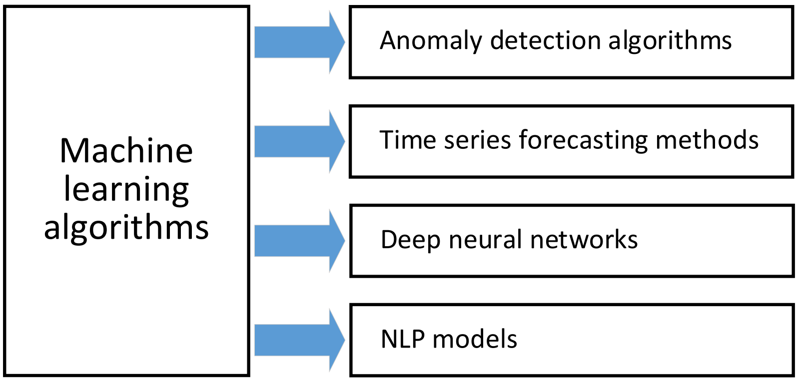

Figure 1. Machine learning algorithms

Anomaly detection algorithms (Isolation Forest, One-Class SVM, Autoencoders) are used to analyze logs and detect unusual patterns in system behavior. They help identify potential problems before they lead to failure. Time series forecasting methods (LSTM, Prophet, ARIMA) are used to analyse historical performance metrics and predict possible deviations, such as server overload or load spikes. Deep neural networks are used to classify logs and automatically determine their criticality, significantly reducing technical support operators' workload. NLP models (Natural Language Processing) are used to automatically analyse text messages about incidents, identify key themes and generate recommendations for troubleshooting.

Implementing artificial intelligence to monitor DevOps processes offers a number of important benefits. Firstly, it reduces the average time to detect incidents due to continuous analysis of logs and streaming data, which allows you to quickly identify potential problems and shortens response times. There is also a reduction in false positives as intelligent models analyse events in context, preventing excessive alarm generation. Automating incident classification with natural language processing algorithms enables the grouping of related events and the identification of potential issue causes, making the work of DevOps engineers much easier. In addition, the system provides proactive troubleshooting. Using predictive models, it not only identifies potential problems, but also suggests ways to fix them or even performs automatic corrective actions.

Actual cases of artificial intelligence implementation demonstrate its effectiveness in monitoring and ensuring the reliability of services. In its Site Reliability Engineering concept, Google actively uses machine learning algorithms to analyse logs and automate monitoring, significantly reducing the response time to problems [20]. Netflix uses AI to analyse performance metrics and predict possible failures in its distributed infrastructure, which helps to prevent incidents and increase service stability [21]. Amazon offers the AWS DevOps Guru service, which analyses logs and detects anomalies in the operation of cloud infrastructures, allowing it to quickly eliminate potential problems and increase the efficiency of DevOps processes [22]. Therefore, automating monitoring using artificial intelligence creates new opportunities to enhance the stability of DevOps systems, boost engineers' efficiency, and reduce the likelihood of failures in critical services.

Optimisation of CI/CD processes. The automation and optimization of CI/CD (Continuous Integration/Continuous Deployment) play a crucial role in enhancing the efficiency of DevOps processes. Leveraging artificial intelligence and machine learning technologies can greatly improve the functionality of deployment pipelines, minimize failure risks, and accelerate the release of updates to production. Conventional CI/CD systems typically function based on preset rules, which can be inefficient in complex and fluctuating environments. By integrating intelligent systems, pipeline settings can be dynamically adjusted based on the current workload, code quality, and infrastructure performance.

The primary focus of CI/CD optimization with artificial intelligence includes several crucial aspects designed to enhance the efficiency and automation of software development and deployment workflows (Fig. 2). AI enables the adjustment of CI/CD pipelines to evolving conditions, smartly tests code to detect potential issues, and enhances the deployment process to ensure both speed and reliability. Reinforcement learning algorithms are used to dynamically adjust pipelines, automatically modifying test parameters and selecting the most optimal configurations. By examining past pipeline execution data, we can forecast potential failures and make necessary modifications to the deployment process. Furthermore, artificial intelligence enables automatic resource allocation based on load levels and the number of active deployments, enhancing overall process efficiency. Smart code testing is a crucial aspect of CI/CD optimization through the use of artificial intelligence. Machine learning algorithms enable the evaluation of code quality and the prediction of potential defects by examining past changes. AI also assists in automatically pinpointing the most crucial tests, thereby reducing test execution time while maintaining effectiveness.

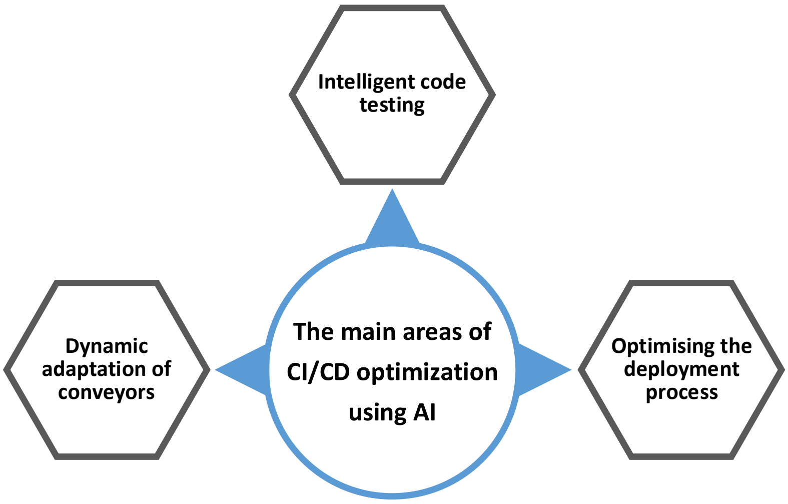

Fig. 2. Main directions of CI/CD optimisation with the help of AI

By forecasting the likelihood of test failure, test scenarios can be pre-optimized to enhance reliability. Optimising the deployment process with AI involves several important approaches. Deep learning is employed to assess release risks and suggest the best timing and approach for implementation. Based on statistical data on system stability and user activity, AI automatically determines the best time for a release. Furthermore, forecasting the chances of a rollback helps mitigate risks during the deployment of new software versions.

The use of artificial intelligence in CI/CD offers considerable advantages for developers in practice. Tools such as GitHub Copilot [23] and Tabnine [24] use machine learning algorithms to analyse code and provide recommendations for improvement at the writing stage. Jenkins AI Plugins [25] offer the ability to analyse test logs, predict errors, and optimise task execution, increasing the efficiency and stability of processes.

Introducing artificial intelligence into CI/CD processes significantly increases the efficiency of DevOps, allowing for automatic adaptation of pipelines, minimising the risk of failures, and optimising resource utilisation. Future studies could focus on enhancing AI optimization capabilities and incorporating self-learning models into cloud-based CI/CD platforms.

Load forecasting and resource management. Efficient resource management is critical for the stability and performance of modern DevOps processes. With the dynamic load on servers and cloud infrastructures, it is necessary not only to respond to changes but also to anticipate them to adapt resources in advance.

Artificial intelligence and machine learning enable the development of self-adapting resource management systems that can forecast workloads, identify optimal infrastructure setups, and autonomously scale resources.

Load forecasting methods play a key role in ensuring the stability and efficiency of systems. In particular, they allow for predicting load changes based on historical data, analysing user requests, and optimising resource allocation. Contemporary methods make extensive use of time-series forecasting, query clustering, deep neural networks, and reinforcement learning algorithms (Fig. 3), offering precise predictions and adaptive resource management in ever-changing environments.

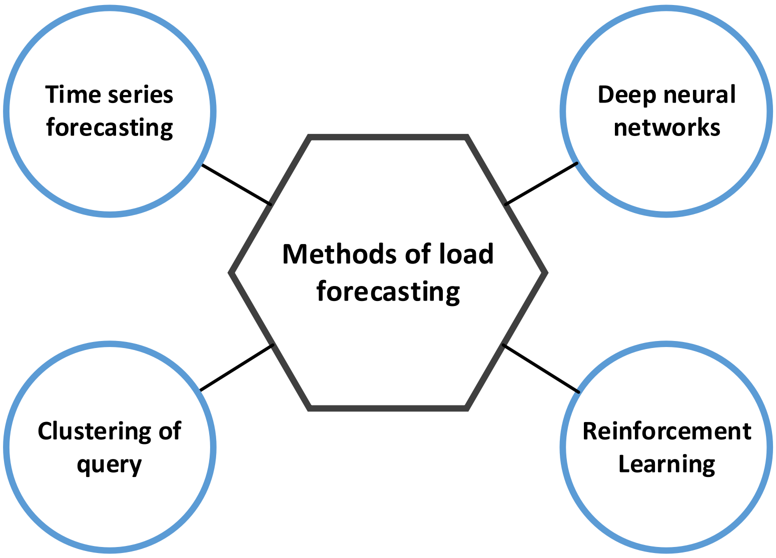

Fig. 3. Methods of load forecasting

Machine learning algorithms are used in load forecasting techniques to analyze data and anticipate fluctuations in system performance. Time series forecastings, such as LSTM, ARIMA, or Prophet, allow you to analyse historical data and identify trends and seasonality, which helps to predict load peaks. Query clustering using K-Means or DBSCAN algorithms helps to group similar queries, identify user behavioral patterns, and predict load increases. Deep neural networks analyse complex multi-factor relationships between CPU, RAM, I/O, and network activity to predict potential system overloads.

Reinforcement Learning algorithms enable real-time adaptive resource management, allowing the system to autonomously make optimal scaling choices to maintain stable performance. Load forecasts allow you to implement effective mechanisms for optimising system performance. Autoscaling automatically adjusts computing resources based on forecasted load. In cloud platforms like AWS Auto Scaling or Google Kubernetes Engine, this enables the dynamic addition or removal of virtual machines or containers.

Load balancing is enhanced through smart traffic distribution algorithms that route requests to the servers with the lowest load, minimizing the chance of overburdening specific nodes. Automatic resource allocation allows for forecasting peak load periods and proactively reserving extra resources, ensuring that system performance remains stable. Energy savings are achieved by temporarily shutting down unused servers or reducing their performance during low-load periods, which can significantly reduce cloud computing costs.

Practical applications in DevOps include using machine learning to optimise infrastructure and resources. Netflix uses these technologies to predict peak loads and automatically scale to serve millions of users simultaneously [26]. Google Cloud AI analyses real-time performance metrics, predicts possible server overloads, and allocates resources in advance [27]. Microsoft Azure AutoML uses machine learning algorithms to automatically manage cloud resources based on historical data and user behavioral models [28].

The research results indicate that integrating artificial intelligence into DevOps processes leads to a considerable improvement in infrastructure management, automation of monitoring, and optimization of software deployment. AI speeds up CI/CD processes, improves service stability, and significantly reduces operating costs through intelligent resource management. Table 2 shows the main areas of AI applications in DevOps and the corresponding improvements they provide.

Table 2

Potential benefits of implementing AI in DevOps

Field of application	Improvements	Potential benefits
Monitoring	Automatic anomaly detection	Reduced incident response time
CI/CD management	Optimise deployment processes	Reduced time to test and release
Load balancing	Intelligent resource scaling	Optimisation of infrastructure costs
Failure prediction	Detect problems before they happen	Increased service stability

Conclusions from this study and prospects for further research in this area

Incorporating artificial intelligence and machine learning into DevOps creates new possibilities for enhancing software development, testing, deployment, and operations. Integrating intelligent algorithms allows not only automating key processes but also increasing system efficiency by predicting possible problems and proactively managing resources.

The findings of the study indicate that incorporating artificial intelligence into DevOps greatly enhances automation of monitoring, optimization of CI/CD processes, and resource management. Machine learning-based automated monitoring and anomaly detection can enhance system control quality, decrease false positives, and accelerate incident response. Self-learning systems adapt to changes in service behavior, providing flexible and accurate infrastructure management. Optimization of CI/CD processes with the help of intelligent algorithms reduces the time for deployment and testing of releases. Dynamic pipeline adjustments and automatic selection of test scenarios help minimize errors. Reinforcement learning techniques enable the system to enhance update releases by learning from past experiences. Load forecasting and intelligent resource management ensure accurate scaling of the server infrastructure per current needs. This helps to avoid overloads, optimize cloud computing costs, and provide an even distribution of requests to improve service performance.

Prospects for further research in the field of artificial intelligence application in DevOps cover several key areas. The creation of AI-powered integrated platforms seeks to develop all-in-one solutions that incorporate machine learning for monitoring, predicting failures, automating CI/CD processes, and optimizing resource management. The expanded use of generative AI opens up opportunities for automating scripting, code optimization, and DevOps infrastructure management with technologies like ChatGPT and Copilot. Combining traditional DevOps approaches with AI-driven SRE (Site Reliability Engineering) will allow the integration of SRE methodologies with intelligent algorithms to improve service availability, reliability, and resilience. The advancement of autonomous DevOps agents can enable the creation of self-learning systems that can independently handle the development, testing, deployment, and monitoring processes without the need for direct human involvement.

Particular attention should be paid to the ethical and security aspects of using AI in DevOps. Exploring the potential risks associated with automated solutions, cybersecurity threats, and the need to ensure transparency and control over algorithmic decisions is crucial.

References

1. Liang P., Zhang H., Wong D., et al. Enhancing Security in DevOps by Integrating Artificial Intelligence and Machine Learning // Journal of Theory and Practice of Engineering Science. – 2024. – Vol. 4, No. 2. – P. 31–37. DOI: 10.53469/jtpes.2024.04(02).05

2. Ali M. S., Puri D. Optimizing DevOps Methodologies with the Integration of Artificial Intelligence // Proceedings of the 2024 3rd International Conference for Innovation in Technology (INOCON). – IEEE, 2024. – P. 1–5. – DOI: 10.1109/INOCON60754.2024.10511490.

3. Abbas S. I., Garg A. AIOps in DevOps: Leveraging Artificial Intelligence for Operations and Monitoring // Proceedings of the 2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL). – IEEE, 2024. – P. 64–70. – DOI: 10.1109/ICSADL61749.2024.00016.

4. Chowdary V. H. D., Reddy K. S., Patel M., et al. DevOps 2.0: Embracing AI/ML, Cloud-Native Development, and a Culture of Continuous Transformation // Proceedings of the 2024 4th International Conference on Pervasive Computing and Social Networking (ICPCSN). – IEEE, 2024. – P. 673–679. – DOI: 10.1109/ICPCSN62568.2024.00112.

5. Shah P. S. U., Ahmad N., Beg M. O. Towards MLOps: A DevOps Tools Recommender System for Machine Learning System // arXiv preprint, arXiv:2402.12867. – 2024. – DOI: 10.48550/arXiv.2402.12867.

6. Pattanayak S., Murthy P., Mehra A. Integrating AI into DevOps Pipelines: Continuous Integration, Continuous Delivery, and Automation in Infrastructural Management: Projections for Future. – 2024. – DOI: 10.30574/ijsra.2024.13.1.1838.

7. Kolawole I., Fakokunde A. Machine Learning Algorithms in DevOps: Optimizing Software Development and Deployment Workflows with Precision // International Journal of Research Publication and Reviews. – Vol. 2582. – P. 7421. – [Electronic resource]. – Available from: https://www.researchgate.net/publication/387705032.

8. Karamitsos I., Albarhami S., Apostolopoulos C. Applying DevOps Practices of Continuous Automation for Machine Learning // Information. – 2020. – Vol. 11, No. 7. – P. 363. – DOI: 10.3390/info11070363.

9. Tatineni S., Chakilam N. V. Integrating Artificial Intelligence with DevOps for Intelligent Infrastructure Management: Optimizing Resource Allocation and Performance in Cloud-Native Applications // Journal of Bioinformatics and Artificial Intelligence. – 2024. – Vol. 4, No. 1. – P. 109–142. – [Electronic resource]. – Available from: https://biotechjournal.org/index.php/jbai/article/view/68.

10. Thompson A. Implementing Scalable DevOps Pipelines for Machine Learning Model Monitoring and Performance Management // Journal of Artificial Intelligence Research. – Sep. 2024. – Vol. 4, No. 2. – P. 117–122. – [Electronic resource]. – Available from: https://www.thesciencebrigade.com/JAIR/article/view/415.

11. Carter M. Scaling DevOps Practices for Distributed Machine Learning: Addressing Challenges in Large-Scale MLOps Deployments // Distributed Learning and Broad Applications in Scientific Research. – 2024. – Vol. 10. – P. 353–359. – [Electronic resource]. – Available from: https://dlabi.org/index.php/journal/article/view/162.

12. Subramanya R., Sierla S., Vyatkin V. From DevOps to MLOps: Overview and Application to Electricity Market Forecasting // Applied Sciences. – 2022. – Vol. 12. – P. 9851. – DOI: https://doi.org/10.3390/app12199851.

13. Bokkena B. Optimizing Cloud Infrastructure Management Using Large Language Models: A DevOps Perspective // Proceedings of the 2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS). – IEEE, 2024. – P. 1401–1406. – DOI: 10.1109/ICSSAS64001.2024.10760725.

14. Anbalagan K. Cloud DevOps and Generative AI: Revolutionizing Software Development and Operations // International Journal of Innovative Research in Science, Engineering and Technology. – Vol. 13, Issue 8, Aug. 2024. – P. 15172–15181. – DOI: 10.15680/IJIRSET.2024.1308130.

15. Mekić E. S., et al. Enhancing Educational Efficiency: Generative AI Chatbots and DevOps in Education 4.0 // Computer Applications in Engineering Education. – 2024. – Vol. 32, No. 6. – P. e22804. – DOI: https://doi.org/10.1002/cae.22804.

16. Gokarna M., Singh R. DevOps: A Historical Review and Future Works // Proceedings of the 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). – IEEE, 2021. – P. 366–371. – DOI: 10.1109/ICCCIS51004.2021.9397235.

17. Faustino J., et al. DevOps Benefits: A Systematic Literature Review // Software: Practice and Experience. – 2022. – Vol. 52, No. 9. – P. 1905–1926. – DOI: https://doi.org/10.1002/spe.3096.

18. Azad N., Hyrynsalmi S. DevOps Critical Success Factors – A Systematic Literature Review // Information and Software Technology. – 2023. – Vol. 157. – P. 107150. – DOI: https://doi.org/10.1016/j.infsof.2023.107150.

19. Zavala E., Franch X., Marco J. Adaptive Monitoring: A Systematic Mapping // Information and Software Technology. – 2019. – Vol. 105. – P. 161–189. – DOI: https://doi.org/10.1016/j.infsof.2018.08.013.

20. What is Site Reliability Engineering (SRE)? [Electronic resource]. Access: https://sre.google/

21. How Netflix Uses Artificial Intelligence [Electronic resource]. – Available from: https://www.argoid.ai/blog/netflix-ai.

22. Amazon DevOps Guru Features [Electronic resource]. – Available from: https://aws.amazon.com/pt/devops-guru/features/.

23. Zhang B., et al. Practices and Challenges of Using GitHub Copilot: An Empirical Study // arXiv preprint, arXiv:2303.08733. – 2023. – DOI: https://doi.org/10.48550/arXiv.2303.08733.

24. Qianyi Y. Systematic Evaluation of AI-Generated Python Code: A Comparative Study Across Progressive Programming Tasks. – 2024. – DOI: https://doi.org/10.21203/rs.3.rs-4955982/v1.

25. Jenkins and Machine Learning Plugins for Data Science [Electronic resource]. – Available from: https://www.jenkins.io/projects/gsoc/2020/project-ideas/machine-learning/.

26. How Data Science Helps Power Worldwide Delivery of Netflix Content [Electronic resource]. – Available from: https://netflixtechblog.com/how-data-science-helps-power-worldwide-delivery-of-netflix-content-bac55800f9a7.

27. Introduction to AI/ML Services in Google Cloud Platform (GCP) [Electronic resource]. – Available from: https://k21academy.com/ai-ml/google/gcp-ai-ml-services/.

28. Buuri J., et al. An Empirical Study Using Microsoft Azure Auto Machine Learning to Detect Zero-Day Attacks // Proceedings of the 25th Annual Conference on Information Technology Education. – 2024. – P. 7–11. – DOI: https://doi.org/10.1145/3686852.368686.