Journal on Electronic and Automation Engineering

Vol: 2(4), December 2023

REST Publisher; ISSN: 2583-6951 (Online)

Website: https://restpublisher.com/journals/jeae/

DOI: https://doi.org/10.46632/jeae/2/4/8

Continual Improvement of Automation in Infrastructural Management Using Devops with Artificial Intelligence Approach

***P. Ramya, D. Sujeetha, S. Prem Kishore, Arul Mohan Raja**

Mahendra Engineering College, Namakkal, Tamil Nadu, India.

*Corresponding author: ramyapb@mahendra.info

Abstract: *In the industry of software development, DevOps play a crucial role for automation, collaboration promotion as well as continuous delivery. The crucial disruptions have done through Artificial Intelligence (AI) for the application of software implement and delivery while integrating with DevOps. The combination of AI as Machine Learning (ML) towards DevOps practices has revolutionized software implement as well as maintenance have created a concrete method for high resilience, efficient as well as scalable method. AI has transformed the method for deploying DevOps. The implementation of AI is considered for improving the process of software development as well as streamlined operations. Traditional DevOps, characterized by Continuous Integration (CI) and Continuous Delivery (CD), often struggles with scalability, error-prone processes, and the need for constant human oversight. By leveraging AI/ML, organizations can predict deployment outcomes, identify potential issues in real time, and automatically rectify them, reducing downtime and enhancing overall system performance. The paper even focuses towards the future of AI in DevOps by introducing the recommendation with envisioning a fully autonomous deployment and maintenance ecosystem. By addressing current challenges and embracing AI/ML technologies, organizations can significantly improve their DevOps processes, leading to faster, more reliable software delivery.*

Keywords: *Automation, continuous integration, artificial intelligence, continuous delivery, DevOps*

1. INTRODUCTION

DevOps is a portmanteau of "development" as well as "operations," representing cultural and operational shift in the software development lifecycle. It consists of communication, collaboration, and integration among software developers (Dev) as well as IT operations (Ops) teams [1]. Core idea behind DevOps is to break down the traditional silos between these two critical functions, enabling a more seamless, efficient, and rapid software delivery process. DevOps practices revolve around several key principles aimed at streamlining the development, deployment, and maintenance of software applications. Continuous Integration (CI)/CD is the backbone of modern DevOps practices. CI encourages developers in integrating code towards shared repository with several times a day. Generating automation and testing have triggered in every integration that allowing teams to detect and address defects early. This approach reduces the complexity of merging code changes and minimizes integration issues can occur while several developers execute with similar codebase. CD has extends CI by automating the release process. Once the transients automated tests done, the code automatically deploy to performance environment and, in some cases, directly to production [2]. This automation ensure about the code present in a deployable state, enabling frequent and dependable release. The result is a important reduction in time-to-market that permit organizations in responding rapidly with respect to market demands as well as customer requirements. The CI/CD pipeline is crucial for maintaining high-quality software and reducing the risk of errors in production. By automation of process deployment and testing, CI/CD minimizes manual interventions, reduces human errors, and accelerates the release cycle. This leads to faster feedback loops, allowing teams to iterate quickly and continuously improve their applications [3].

As DevOps practices have matured, the complexity of managing and optimizing CI/CD pipelines, infrastructure, and application performance has increased. To address these challenges, organizations have increases turning to AI as well as ML technologies to enhance DevOps processes. However, AI has refers to human intelligence simulation processes through machines, especially in computer systems. Hence, the AI envelopes a broad technique ranges including ML, computer vision, Natural Language Processing (NLP), etc. (Tatineni & Chinamanagonda, 2022). ML is a subset of AI that focuses algorithm development which permits computers in learning from as well as decision making with respect to data. In the DevOps context, AI technologies can be leveraged in automating and optimize several aspects of the Software Development Life Cycle (SDLC). These technologies can analyze vast amounts of data generated by CI/CD pipelines, infrastructure, and applications, enabling intelligent decision-making and proactive issue resolution [4].

DevOps major goal has streamlined the SDLC by promoting automation, collaboration, and iterative processes. From the heart of traditional DevOps practices involves CI and CD pipelines that automate the integrating code process that modifies in testing and delivering it to production environments. CI involves frequent code modification merging from many developers towards shared repository which is many times a day. Every integration has triggered an automated building process while the code gets compiled, and unit tests have executed. CI focuses in detecting and addressing integration problem earlier as well as reducing the conflicts and bugs risk in the consequent stage development. CD even extends the CI process by the code deployment process to production environments. Once code passes all automated tests, it automatically deployed for production-like environment, whereas subsequent testing and validation can occur. This makes certain code is a frequent deployable state that allowed for quick and reliable releases. Some organizations take this a step further with Continuous Deployment, where each modification passes all pipeline stages gets automatically deployed to production. CI/CD pipelines enable organizations to achieve faster, more reliable software delivery by automating critical processes and reducing the time between code changes and their deployment to production [5].

The paper's organization captures all subject aspects to ensure the reader understands the topic well. It will include a brief look at the historical background of DevOps and the place of automation in it, an elaborated examination of automation through the use of artificial intelligence, as well as an evaluation of the cloud's importance, which is a focal constituent of the topic. As described in the paper, AI has enhanced continuous integration deployment, and this paper will bring out the challenges and issues that may arise when implementing these technologies. In conclusion, demonstrations of actual life applications and some case studies will be made, and the future contour and viable opportunities for this field will also be discussed. In this regard, this paper will outline a procedure to show how AI and cloud computing have revolutionized the DevOps process and a guideline for a firm that seeks to adopt the technology.

2. LITERATURE REVIEW

The most usual method in using AI with DevOps is for both CI and CD. AI assist in automating the building, deploying and testing process code that have modified the pass suitable testing has integrating towards existing codebase as well as deploying in production environment at once. This process has assisted in reducing the risk error as well as improving the complete quality of the software that developed. Assortment of AI with infrastructure advancement has safety, sustainability and effective innovation. Based on the urbanization rate and demands to smart infrastructure, the AI has defined unique methods in solving these problems.

Automated code quality analysis using AI/ML involves leveraging intelligent algorithms to assess and improve the quality of code. AI/ML models can analyze codebases to identify potential issues, enforce coding standards, and suggest improvements. AI/ML models can predict and address code issues by analyzing historical code changes, identifying patterns associated with bugs, and learning from previous code reviews [6]. These models have been trained for large code datasets and enable it for recognizing an usual coding errors as well as potential vulnerabilities. For example, AI models can predict areas of the code that are more likely to introduce bugs based on patterns observed in similar codebases. They can also suggest fixes or improvements by comparing the code against best practices and known solutions. This proactive approach assist in identifying issues earlier in the development process and minimize the likelihood of bugs that reaches production. DeepCode uses ML algorithms to analyze code as well as providing insights into potential issues. It leverages a large dataset of open-source projects in learning coding patterns and detect problems such as security vulnerabilities, performance issues, and code smells. Codacy integrates with CI/CD pipelines to provide automated code reviews [7]. It uses AI to evaluate code quality, detect issues, and enforce coding standards. Codacy supports multiple programming languages and offers customizable rules and integrations with

popular version control systems. These tools integrate seamlessly towards the development workflow, provide real-time feedback as well as recommendations to improve code quality.

Case study, Facebook uses predictive analytics to optimize its deployment process. By analyzing historical deployment data, the company can predict the likelihood of deployment failures and adjust its deployment strategies accordingly. This approach has helped Facebook maintain high availability and performance during updates [8]. Netflix employs AI-driven predictive analytics to manage its complex deployment environment. The company uses machine learning models to forecast deployment success rates and recommend deployment strategies. This has enabled Netflix to deploy code changes with minimal disruption and maintain a high level of service availability. Anomaly detection in CI/CD pipelines involves using AI/ML models to monitor and identify unusual patterns or behaviors in the pipeline. This allows for early detection of issues and automated responses to prevent disruptions. AI/ML models continuously monitor CI/CD pipelines for anomalies, such as unexpected changes in build times, test failures, or deployment errors. These models have analyzed historical data and learn normal operating patterns, enabling them to detect deviations in real-time [9]. Case Study, GitHub Actions integrates AI-driven anomaly detection into its CI/CD pipelines. The platform uses machine learning models to monitor workflows for unusual behavior and automatically takes corrective actions, such as halting a problematic deployment or notifying the team of potential issues. Self-healing infrastructure refers to systems which automatically detecting and recovering from failures with no human intervention. AI-driven self healing systems use machine learning algorithms to identify and address issues, ensuring continuous operation and minimal downtime. Self-healing systems leverage AI/ML models to monitor infrastructure health and performance [10].

The platform uses machine learning models to detect and address issues in real-time, such as automatically scaling resources in response to increased demand or recovering from service failures. Amazon Web Services (AWS) offers self-healing capabilities through its services like AWS Auto Scaling and AWS Elastic Load Balancing. These services use AI/ML to monitor infrastructure and automatically adjust resources or redirect traffic in response to failures or changes in demand [11]. An open-source ML framework has developed through Google, whereas TensorFlow has widely utilized to build and deploy AI models. This provides complete ecosystem for training and serving ML models, making it suitable for various DevOps applications, including predictive analytics and anomaly detection. Developed via Facebook, PyTorch is subsequent familiar ML framework is recognized its dynamic computational graph as well as robust in usage [12]. PyTorch is used for developing and deploying ML models in DevOps, particularly for tasks such as code quality analysis and real-time monitoring.

AI/ML frameworks can be integrated with existing DevOps tools to enhance their capabilities.
For example: Jenkins is a familiar CI and CD tool which can extended with AI/ML plugins to enable intelligent code analysis, predictive analytics, and anomaly detection within CI/CD pipelines. Docker, a containerization platform, can leverage AI/ML models to optimize resource allocation, manage container health, and automate scaling decisions [13]. Terraform, an IaC tool, can integrate with AI-driven resource management platforms to optimize infrastructure provisioning and configuration. Several platforms offer AI-driven capabilities specifically designed for DevOps environments. These platforms provide integrated solutions for monitoring, automation, and optimization. Kubernetes, an open-source container orchestration platform, can integrate with AI/ML tools to enhance its functionality. For an instance, AI-driven algorithms have optimized resource allocation, manage container health, and automate scaling decisions based on real-time data. Datadog provides a comprehensive observability platform with AI-driven features for monitoring, logging, and analytics. Its machine learning capabilities enable anomaly detection, predictive analytics, and automated incident response [14].

LinkedIn uses AI/ML for various DevOps applications, including automated code quality analysis and predictive analytics for deployments. The company leverages machine learning models to improve code reviews, optimize deployment strategies, and ensure system reliability. IBM integrates AI/ML into its DevOps tools to enhance automation and monitoring. For example, IBM's Watson AIOps platform uses machine learning to analyze system logs, detect anomalies, and automate incident management, improving operational efficiency and reducing downtime. Spotify employs AI-driven tools for monitoring and managing its complex infrastructure [15]. The company uses machine learning models to analyze performance metrics, detect anomalies, and optimize resource usage, ensuring a seamless user experience. The effectiveness of AI models in DevOps largely depends on the quality and availability of the data they are trained on. High-quality data is crucial for developing accurate and reliable models, but several challenges can arise in this regard: Data used for training AI/ML models may come from multiple sources and may not always be consistent.

Inconsistent data can lead to models that perform poorly or provide unreliable results. Incomplete data can limit the ability of AI/ML models to learn effectively [16]. Difficulty in Integrating AI/ML into Existing DevOps Pipelines, AI/ML tools and frameworks may require specialized infrastructure and resources, such as GPU or TPU acceleration. Integrating these requirements with existing DevOps infrastructure can be complex and may necessitate significant changes. DevOps pipelines often involve a diverse set of tools for CI/CD, monitoring, and deployment. Integrating AI/ML models into these tools can be challenging, especially if the tools do not natively support AI/ML capabilities. Implementing AI/ML solutions requires specialized skills and knowledge. DevOps teams may need to acquire new expertise or collaborate with data scientists to effectively integrate and manage AI/ML models. Deploying AI/ML models into production environments requires careful planning and management. This includes managing model versions, handling updates, and ensuring that models perform as expected in real-world conditions [17].

3. RESEARCH METHODOLOGY

AI can also be utilized to aid CI initiatives in DevOps organizations. AI can discover patterns and trends in data from many sources, such as logs, performance metrics, and user feedback, and utilize it to suggest areas for development. This data may be used later to drive subsequent development efforts and improve the software delivery process. The integration of AI/ML is expected to drive even greater levels of automation in DevOps processes. This includes automating more complex tasks, such as intelligent incident response, autonomous infrastructure management, and advanced code optimization. AI/ML models will continue to advance in their ability to predict system behavior, performance issues, and potential failures. This will enable more proactive and precise decision-making, reducing the likelihood of disruptions and improving system reliability. As edge computing becomes more prevalent, AI/ML will play a key role in managing and optimizing edge deployments. AI-driven tools will help manage distributed systems, optimize resource allocation, and enhance real-time decision-making at the edge. This research work illustrates the benefits of AI in DevOps process that majorly concentrated on CI, CD as well as automated infrastructure. This even discussed about individual AI tools as well as techniques with their current issues, advantages and prospects in using AI with DevOps.

AI-Powered CI

One of the pillars is CI that represents the developers with CI code from developers in major branch. Once the development and testing is done, the development is executed successfully by containerized as well as releases candidate have signed.

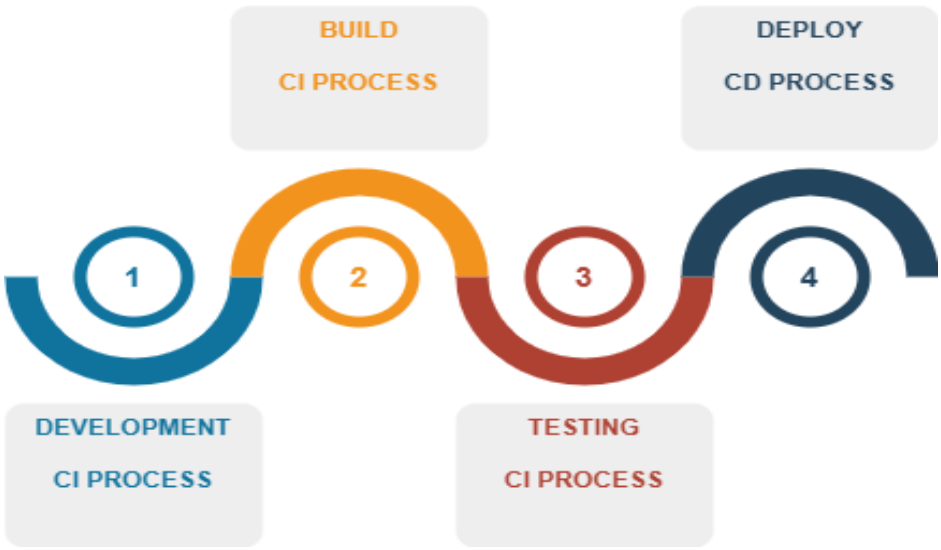

FIGURE 1. Continuous improvement of CI/CD process

Three major CI objectives are

1. To identify problem at earliest opportunities.

2. To prevent integration problems and also providing good quality.
3. To secure release candidates.

Moreover, the improvement of each stages in CI and CD integration can be executed by AI are as follow

Code Branching
In case of large teams, they are facing problems in maintaining several branching of codes in DevOps like tracking the issues and merging conflicts.
AI tool with solution: IntelliCode is a tool from Microsoft has provided developers with AI assistance for powerful usage of branch management. It also assists in finding the best branch strategies for minimizing or avoiding confusions.

Merges
When working with large and complicated projects, there are certain conflicts get created with some code changes.
AI tool with solution: DeepCode tool and Sider tool are the AI tools for scanning and analyzing code before it gets merged for avoiding merge conflicts. This tool has made an alert to developers about similar merge conflict and even provides solution before it occurs by understanding the code's pattern and structure of changes.

Code Commits
The commits concepts is considered to be best in managing codebase but cause problematic due to confusion creation as well as employing distinct policies frequently.
AI tool with solution: Kite is the tool used as AI coding assistant that assist in code improvement before changes is committed. This tool help in understanding the history of code's commits as well as recommended the best practices. It even provides feedback in code modification over real time before the code commits.

Static Scanning
The DevOps may be annoying as well as time consuming and that may lead to false results.
AI tool with solution: DeepCode tool is used to scan by code and it assist in identifying issues in the code as provide feasible solutions for the problems with respect to earlier scan. It assist in improving the tools of static analysis and understand from previous scan in analyzing accurate and providing corrective solution for the identified issues.

Looking for Quality Errors and Security Issues
It is a slow process and influences the flood to teams with potential problem in facing.
AI tool with solution: ShiftLeft tool is used for security risks as well as employing AI for filtering by risks it identifies. This tool assist in improving the scan operation efficiency and speed to find out the issues available in earlier scan as well as prioritizing based on severity.

Builds
The project has experienced build failure because of various reasoning like wrong configuration and poor quality code.
AI tool with solution: CircleCI Insight has offered successive prediction in building code using ML that assist in preventing failure. With the help of past data as well as pattern, AI has predicted build failure whereas it assists in automating the build process as well as handling configuration to avoid mistakes.

Unit Testing
Set of unit tests have been created and updated need more effort as well as time. These tests require more extensive.
AI tool with solution: Diffblue Cover's tool is used for generating unit tests by AI for increasing speed and coverage. Based on the code analysis, AI has assist in creating unit tests for high efficient in coverage than traditional tests and execute robust.

Functional Integration Testing
All tests are demanding with respect to the resource as well as time need to perform it.
AI tool with solution: Test.AI tool is used for producing as well as performing functional tests and reduces the requirement in developing test cases. It generate functional integration test with respect to the system behavior that desire minimum maintenance efforts.

Containerization

The unplanned configurations of containers cause application failure as well as several security breaches.
AI tool with solution: Magalix tool has utilized historical data as well as the present requirement for defining ideal containers as well as highlighting potential gets configurations. It even improves as well as optimizing the container configuration process.

4. RESULT AND DISCUSSION

The advantage of AI-powered with Infrastructure as Code (IaC) has generated DevOps is as follows

Faster Infrastructure Provisioning: Due to setup and configuration are performed automatically with AI, the IaC deployment time is significantly reduced. This rapid deployment allows developers to start exploring novel settings rapidly, reducing the time required for introducing products to market.

Improved Infrastructure Security: AI's success stems from its ability to discover flaws in settings while comparing them with security requirements. It also helps to eliminate dangers when possible attackers attempt to capitalize them as well.

Reduced Operational Costs: AI-powered IaC may decrease operational costs by reducing the sum of time and assets required for repetitive tasks. Professionals in organizations may utilize resources more efficiently and eliminate unnecessary expenditures.

Improved Consistency and Standardization: AI guarantees that the system configurations are consistent across different environments. This uniformity reduces the likelihood of inaccuracies and mistakes which enhances system accuracy.

Predictive Resource Management: AI can also monitor customers' usage patterns and predict the required resources for the organization's growth so that it can plan its infrastructure accordingly. This capability assists in sustaining performance while at the same time reducing costs.
Simplified Maintenance and Updates: AI facilitates infrastructure management and updates by automating patching as well as configuration changes. This will assist in lowering the time consumption and the number of errors occurs while performing modifications.

Better Compliance and Auditability: Based on findings, IaC can use AI in verifying the set corresponds with the regulations as well as organizational policies of diverse businesses. This capability simplifies audits and ensures that infrastructure meet the needed standards.

Enhanced Collaboration: AI IaC is code-based that assists in achieving the goals of the culture of DevOps by linking the development as well as operational teams.

Rapid Testing and Validation: AI may also assist with autonomous testing and validation of infrastructural code to ensure that configurations perform according to expectation before deployment. This reduces the risk of difficulties emerging during production.

Dynamic Infrastructure Optimization: AI may perform immediate changes to infrastructure resources by continuously analyzing performance and enhancing the performance of existing systems.

Here, intelligent deployment strategies are listed as one of the main ways AI contributes to continuous deployment. Some of the traditional approaches to deployment are based on a set schedule and involve predefined steps and actions; some examples are blue-green deployments or canary releases, where the change is applied incrementally so the negative impact, if any, would be minimized. Nevertheless, these strategies can be helpful; however, their gains are usually fixed and may not respond to changes that characterize production environments. AI improves these strategies by having constant data feed from the actual production setting on issues like user interactions, system performances, and resource consumption, among others, in defining the deployment strategy. For instance, it can slow down or stop the implementation of the first phase of a canary release if it identifies emerging problems with realized performance. This dynamic calibration minimizes multiple failures and guarantees that changes are delivered securely and in the best way.

There is also, however, the use of AI in the CD process, and this AI is mainly used to make decisions, especially rollback decisions. In continuous deployment, if a new version of the application is deployed, problems arise, but it takes a short time to do so. In the past, this entails conducting manual monitoring and intervention, which, even though time-consuming, can also involve some errors. AI, on the other hand, can constantly monitor the KPIs of the business/ organization and notice a potential concern with new deployment suggested by some of the KPIs. Also, based on the severity of the issue, the AI system can decide to go back to the previously stable state of the application, hence minimizing the time the users are affected by the problem. Apart from raising the speed of working cases, this capability also ensures that only pure, data-driven, and up-to-date rollback decisions can be made.

Still, on resource utilization in deployment, another area that has received a boost in continuous deployment through the incorporation of AI is this. This is usually coupled with the fact that when new versions of these applications are developed, there is typically the need to continually poll for the CPU, memory, and network bandwidth that the applications use to be sure that they are not impairing the system. AI can approximate the resources needed for a fresh deployment in resource management by assessing the data gleaned from prior experiences and synchronizing the latest situations. However, AI still needs to assign more resources for a new deployment. Therefore, optimizing this infrastructure would allow depreciation on the costs that go with it and enhance the deployment of this infrastructure with little or no hitches on performance.

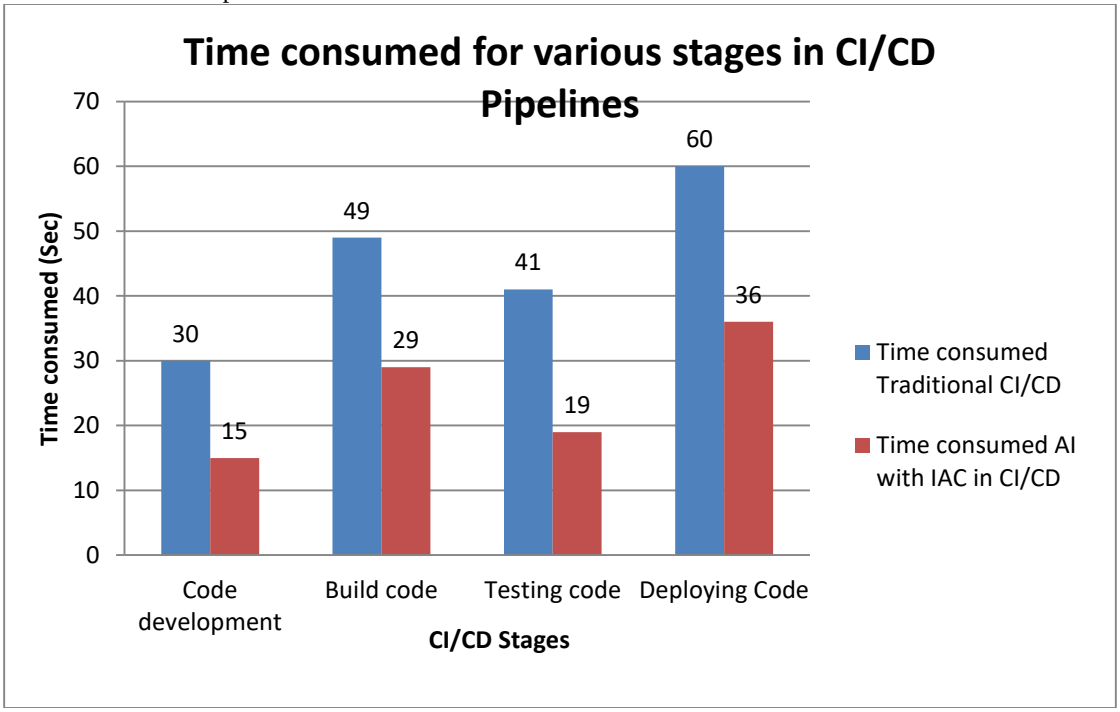

FIGURE 2. Comparison of performance in time consumed for various stage in CI/CD pipeline

Figure 2 illustrates performance of time consumed for various stages in which AI with IAC has better deployment of code with less time consumption with better efficiency.

5. CONCLUSION

DevOps AI is a relatively new concept that has been embraced to boost conventional application development and management methods. By incorporating AI into DevOps practices, organizations experience what is now effectively a revolution in development and operation activities and processes. AI as a tool and technology makes it possible to automate the manual and repetitive parts of the job, utilize resources, and manage the qualities of assurance, thus

playing a role in optimizing new and advanced software delivery life cycles. As is evidenced by practical examples, businesses in most sectors have started harnessing the capabilities of AI in DevOps. The case of Netflix, which utilizes AI for monitoring and performance optimization to LinkedIn's consolidated intelligent code review and deployment processes, perfectly articulates the benefits that AI has brought about in enhancing software quality and operational efficiency. Similarly, AI assumes testing, security, and optimization responsibilities in finance, aerospace, and retail to respond to complex issues and facilitate faster adaptability and business continuity.

REFERENCES

[1]. Tatineni, S., & Chinamanagonda, S. (2021). Leveraging artificial intelligence for predictive analytics in devops: enhancing continuous integration and continuous deployment pipelines for optimal performance. Journal of Artificial Intelligence Research and Applications, 1(1), 103-138.

[2]. Tatineni, S., & Katari, A. (2021). Advanced AI-Driven techniques for integrating DevOps and MLOps: enhancing continuous integration, deployment, and monitoring in machine learning projects. Journal of Science & Technology, 2(2), 68-98.

[3]. Pakalapati, N., Konidena, B. K., & Mohamed, I. A. (2023). Unlocking the power of AI/ML in DevSecOps: Strategies and best practices. Journal of Knowledge Learning and Science Technology, 2(2), 176-188.

[4]. Tatineni, S., & Chinamanagonda, S. (2022). Machine Learning Operations (MLOps) and DevOps integration with artificial intelligence: techniques for automated model deployment and management. Journal of Artificial Intelligence Research, 2(1), 47-81.

[5]. Rosch-Grace, D., & Straub, J. (2022). Analysis of the likelihood of quantum computing proliferation. Technology in Society, 68, 101880.

[6]. Battina, D. S. (2021). Ai and devops in information technology and its future in the united states. International Journal of Creative Research Thoughts (IJCRT).

[7]. Lwakatare, L. E., Crnkovic, I., & Bosch, J. (2020, September). DevOps for AI–Challenges in Development of AI-enabled Applications. In 2020 international conference on software, telecommunications and computer networks (SoftCOM) (pp. 1-6). IEEE.

[8]. Mboweni, T., Masombuka, T., & Dongmo, C. (2022, July). A systematic review of machine learning devops. In 2022 international conference on electrical, computer and energy technologies (ICECET) (pp. 1-6). IEEE.

[9]. Bosch, N., & Bosch, J. (2020, August). Software logs for machine learning in a DevOps environment. In 2020 46th Euromicro Conference on Software Engineering andAdvanced Applications (SEAA) (pp. 29-33). IEEE.

[10]. Kumar, R., Bansal, C., Maddila, C., Sharma, N., Martelock, S., & Bhargava, R. (2019, May). Building sankie: An ai platform for devops. In 2019 IEEE/ACM 1st international workshop on bots in software engineering (BotSE) (pp. 48-53). IEEE.

[11]. Alenezi, M., Zarour, M., & Akour, M. (2022). Can artificial intelligence transform DevOps?. Arxiv Preprint Arxiv:2206.00225.

[12]. Venigandla, K., & Vemuri, N. (2022). Autonomous DevOps: Integrating RPA, AI, and ML for self-optimizing development pipelines. Asian Journal of Multidisciplinary Research & Review, 3(2), 214-231.

[13]. Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps practices of continuous automation for machine learning. Information, 11(7), 363.

[14]. Sen, A. (2021). DevOps, DevSecOps, AIOPS-paradigms to IT operations. In Evolving Technologies for Computing, Communication and Smart World: Proceedings of ETCCS 2020 (pp. 211-221). Springer Singapore.

[15]. Alnafessah, A., Gias, A. U., Wang, R., Zhu, L., Casale, G., & Filieri, A. (2021). Qualityaware devops research: Where do we stand?. IEEE Access, 9, 44476-44489.

[16]. Pelluru, K. (2023). Advancing software development in 2023: the convergence of MLOps and DevOps. Advances in Computer Sciences, 6(1), 1-14.

[17]. Eramo, R., Muttillo, V., Berardinelli, L., Bruneliere, H., Gomez, A., Bagnato, A., ... & Cicchetti, A. (2021, September). Aidoart: Ai-augmented automation for devops, a model-based framework for continuous development in cyber-physical systems. In 2021 24th Euromicro Conference on Digital System Design (DSD) (pp. 303-310). IEEE.

