NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696
VOLUME 6, ISSUE 3, Mar.-2019

AN INTELLIGENT DEVOPS PLATFORM RESEARCH AND DESIGN BASED ON MACHINE LEARNING

Dhaya Sindhu Battina
Sr. Data Engineer & Department of Information Technology
USA

Abstract
The main purpose of this paper is to review DevOps theoretical framework using the machine learning method. There have been several changes in the intelligent communication and Internet sectors as a result of the continual deepening and extension of IT business based on artificial intelligence, machine learning, and blockchain technologies [1]. Mature IT companies are required to deal with enormous volumes of data as part of their regular DevOps (Development & Operations) tasks. Over time, it became apparent that this data came from a variety of sources, was in a variety of formats, and had other difficulties [1]. DevOps developing computer software and hardware technologies that are both efficient and cost-effective has become a critical activity that must be completed. It's estimated that DevOps accounts for more than half of the SLC. In terms of overall business control, business risk management, and business cost management, it influences the whole IT company. Using machine learning methods to conduct research and design an intelligent DevOps platform, this project aims to increase the efficiency of DevOps engineers while also ensuring that DevOps has a high intelligence level. This project also helps to move DevOps towards informatization by helping engineers analyze large amounts of various system alarms [1].

Keywords: DevOps, intelligent DevOps, Machine learning, software development, automation tools

I. INTRODUCTION

There's no denying that DevOps is a rising star in the software world. In DevOps, orchestration calls for more complex tools and technologies to get the greatest outcomes. DevOps is being transformed and software development techniques are being revolutionized because of the combination of AI and data science [2]. According to Gartner, by 2023, 40% of DevOps teams will use artificial intelligence-enabled AIOps monitoring solutions to supplement their existing application and infrastructure monitoring [2]. By using AI and ML-driven requirements management tools, DevOps teams may save a considerable amount of time, allowing them to devote more of their energy to developing software products on schedule.

To solve difficult real-world issues, revolutionize companies, and provide value, machine learning (ML) is becoming more important in the area of data science. In the scientific and machine learning communities, many data science teams are thus working to increase the entire business value via descriptive and predictive models [2,3]. DevOps concepts are thus studied by data scientists and ML operations engineering teams for their ML systems. The term "DevOps" refers to a collection of software and systems engineering processes and technologies. To put it another way, software engineering is a field that focuses on creating and using advanced software systems. Data science is more about analyzing and deriving insights from data than it is about writing programs. When it comes to an iterative strategy, Agile [4] refers to one that emphasizes teamwork, customer input, and frequent, minor releases. To achieve business goals, the conventional operational and development teams are combined with DevOps and Agile, creating a work environment where operations are continuously improved by a cross-functional team of developers and operators. The strategic goal of DevOps is to find ways to improve service quality and features while also meeting the demands of its customers [5]. To offer an ML pipeline model, ML data scientists and ML operations engineering teams have included manual procedures. Due to the reliance on data collection, preparation and pre-processing, model training, validation, and testing, this technique may generate unexpected findings. This strategy also led to the conclusion that using our manual methodology for ML projects has no obvious advantages. The substantial operating expenditures and delays caused by this ML manual pipeline approach have a negative impact on the company's revenue and quality reputation [5].

It is recommended in this study that two DevOps concepts be included in the MLOps methodology to answer these research problems. Designing an automated machine learning pipeline based on two DevOps [6] principles: continuous integration (CI) and continuous delivery is the primary goal (CD). Practice MLOps, which advocates automation and monitoring throughout the ML system building process, implies that all of these activities are part of the MLOps process and are included in the MLOps practice. Continuous delivery (CD) and continuous integration (CI) are significant DevOps techniques included in the operation of ML complex systems (CI). CI's capabilities extend beyond testing and validating code and components to include data, data schemas, and model tests and validations as well [6]. CD is no longer limited to a single package or service, but rather an ML pipeline that may automatically deploy another ML service upon completion.

II. PROBLEM STATEMENT

The main problem that this paper will address is to review Intelligent DevOps Platform Research and Design Based on Machine Learning. Data security issues, inconsistent testing methods, out-of-date Legacy modernization systems, and a lack of communication are some of the problems businesses encounter while implementing DevOps. One may, however, safely navigate the storm by having clarity on the techniques and underlying concepts you need to follow to remedy the issue(s) [7]. Adopting new technologies to streamline software development, testing, and deployments across diverse departments within a business is the most difficult obstacle to successfully execute DevOps. With AI, DevOps and production cycles may be streamlined, and many aspects of release and DevOps can be addressed. With it, DevOps functions may be accelerated while costs and time to market are reduced [7,8]. As the emphasis shifts from development to delivery, DevOps has become a goal for every firm. To manage the volume of data moving in dynamic application settings, managing a DevOps team necessitates significant work.

III. LITERATURE REVIEW

A. DevOps Model and Practices

Development and operations are referred to as DevOps [8]. It is a methodology that tries to bring together development, quality assurance, and operations into a unified, uninterrupted set of activities to achieve greater efficiency. This technique is a logical consequence of the software development paradigm's Agile and continuous delivery methodologies. DevOps, on the other hand, is more than a collection of techniques. Cross-functional team collaboration is fostered by a company's culture or philosophy. A major advantage of DevOps is that it does not need major technological changes, instead of focusing on how a team collaborates. In DevOps culture, teamwork is essential: the entire success of a process is dependent on it, and there are specific principles and practices that DevOps teams follow to achieve this success.

Using DevOps [9] calls for a continuous delivery cycle that includes planning, development, and testing before being deployed and released with active cooperation between team members. Continuous delivery is a methodology that uses automation to simplify the development, testing, and deployment processes. To make testing easier, programmers commit code in tiny pieces during the development process many times a day. Automated code testing is performed by a quality assurance team. The engineering team is notified whenever defects or vulnerabilities are found. At this point, version control is also required to look for any integration issues ahead of time. Software engineers can keep track of file changes using a version control system (VCS), which they can then distribute to other team members wherever they may be located. Every piece of code that passes an automated test ends up in the same place: a server-side repository. Code releases regularly help to avoid what is known as 'integration hell,' which occurs when the differences between separate code trees and the mainstream code grow so significantly over time that assimilation takes longer than real work. In terms of continuous integration, Jenkins [9] and GitLab CI [9] are the most widely used tools. After that, it's up and running on a live public server. Code must be deployed in such a manner that it does not interfere with already working functionality and may be made accessible to a wide number of people. When new features are often deployed, they may be evaluated and verified early on using a "fail fast" methodology. Engineers can launch a product increment more quickly with the aid of automated technologies. Chef, Puppet, Azure Resource Manager, Google Cloud Deployment Manager, and Amazon S3 are the most widely used. In the last step of the DevOps lifecycle, the focus is on evaluating the whole cycle's effectiveness. By keeping an eye out for any problems in a process, you may report any inconsistencies you find and make improvements to the product as a whole [9,10].

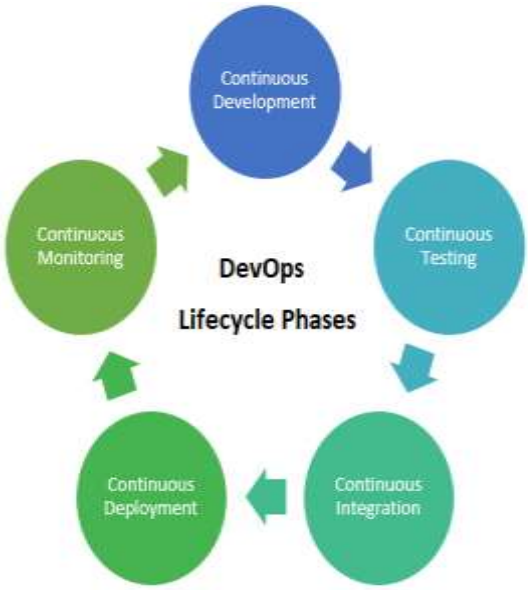

Fig i: DevOps Lifecycle phases

B. Continuous integration (CI) and Continuous Delivery (CD)

With continuous integration (CI), members of a team integrate and merge development work (such as code) regularly, such as many times a day (for example). A shorter and more frequent software release cycle improved software quality and increased productivity are all possible with CI [10]. This involves the creation and testing of software using automated processes. It is the goal of continuous delivery (CD) to ensure that an application is always in a production-ready condition after the successful completion of automated tests and quality checks. CI and deployment automation, for example, is used by CD to provide software in a production-like environment automatically [10]. When it comes to CD, it's all about pushing. According to [10,11], this approach has various advantages, including decreased implementation risk, cheaper costs, and quicker access to user input.

C. MLOps and its Significance

Although its scope and application vary, AIOps and MLOps are both new terms in software development. To maximize the business value of machine learning solutions, MLOps principles are used at every stage of the workflow. To summarize, MLOps combines DevOps with machine learning and delivers a variety of commercial advantages, as detailed in this article. It's become standard practice in software development to use a variety of tools and techniques to stay up with modern application demands [11]. DevOps is becoming more complex to construct quick and efficient applications for modern consumers, and machine learning software is not left out in this trend with the most current buzz around machine learning and artificial intelligence. Machine Learning Operations (MLOps) and Artificial Intelligence for IT operations come to mind when discussing the integration of machine learning or artificial intelligence in DevOps (AIOps). Both phrases have a similar meaning, however, there are a few key differences [11].

To put it another way, MLOps refers to the use of DevOps methods, practices, and tools in machine learning initiatives. Continuous integration and automation (CI/CD) are applied to machine learning projects from design to training and deployment using DevOps methods such as collaboration, source control, testing, and continuous integration (CI/CD) [11]. For example, MLOps can be used to integrate data science and IT operations teams more seamlessly so that machine learning projects can be released with more efficiency and reliability. For a technology sector to fulfill its full potential, it must be combined with structured, business-oriented processes. Otherwise, machine learning risks degrading to the status of a mere experiment and liability for a company, with little or no economic benefits [11]. MLOps is used in the iterative design, model development, and operations phases to maximize the commercial benefits of machine learning.

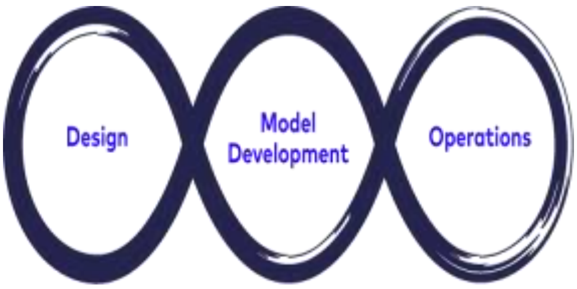

Fig ii: MLOps phases

During the design phase of a machine learning project, the focus is on determining the project's specific business and data requirements. A suitable business model and machine learning solution to handle the user's problem are designed in this phase, which identifies the application's potential users. Also, the software's potential uses are developed, and a check is made to see if the data essential for training an artificial intelligence model is available. The information gleaned from each of these steps is subsequently applied to the ML-powered app's architecture design [11]. In the experimentation and development phase, the product from the preceding design stage is tested to validate the planned ML solution's actual implementation in the real world. The ML method is then iterated upon until a stable model is achieved that can be used in production. To put the model into production, proven DevOps methodologies such as MLOps in Operations are utilized. This ML-powered application phase makes use of testing, monitoring, versioning, automation, and continuous delivery as well as other DevOps procedures [11,12]. To get a machine learning project ready for users, each of the above phases must be employed as a follow-up process to the prior one.

D. The MLOps Principles

MLOps, like DevOps, is guided by principles in both its operation and application. MLOps, on the other hand, is a machine learning version of DevOps. While there may be others, the following are the most critical MLOps principles:

1. Testing

As with DevOps, testing is an essential component of software development and machine learning. Data pipeline, ML model pipeline, and application pipeline make up MLOps, which is a structured testing process for machine learning systems [12]. To verify the accuracy of the ML model, each component is thoroughly evaluated for integration and usability.

2. Monitoring

When an ML model is deployed, it is critical to monitor it regularly to ensure its correctness, as is demonstrated by this principle. Dependency, data version, usage, and model modification are all monitored to make sure that the model functions as planned. To ensure that the model is performing as expected, the desired behaviors should be recorded in advance and used as a benchmark to take corrective action if the model underperforms or spikes sporadically [12].

3. Versioning

For the machine learning model, this means utilizing a version control system to keep track of all the changes made to the data sets and code. For this reason, the version control tool tracks and saves different versions of the machine learning model as events change the data or produce anomalies [12]. When an issue emerges, it's simple to roll back to a previous version or see exactly where a bug was fixed.

4. Continuous "Everything"

MLOps has embraced DevOps' continuous workflow in the ML pipeline, which is one of the benefits of MLOps. A machine learning model is only good for a short period, and it is vulnerable to change based on fresh data. To make ML engineering techniques such as continuous integration (CI), continuous delivery (CD), continual testing (CT), and continual monitoring easier to adopt, MLOps was developed (CM) [12].

5. Automation

Prioritizing automation is critical to the effective implementation of MLOps in any machine learning project. The maturity of the ML process is determined by the level of automation of your ML model, which in turn enhances the pace at which machine learning models can be trained, produced, and deployed. Using important occurrences as triggers for each workflow activity, this theory enables a fully automated machine learning workflow that does not require any human participation. There are three stages to adopting MLOps' automation;

Manual Process:

As a first step, models are manually checked and tested before being executed iteratively to train the model for future automated processes. ML Pipeline Automation: At this point, the model is given continual training to keep it sharp. There is no need for manual interaction at this point because the validation and retraining processes are automatically started as fresh data become available [12].

CI/CD Pipeline Automation:

DevOps' continuous integration and delivery (CI/CD) pipeline are used in the third phase to automatically develop, test, and deploy machine learning models. DevOps' continuous integration and delivery (CI/CD) pipeline are used in the third phase to automatically develop, test, and deploy machine learning models.

6. Reproducibility

Keeping MLOps reproducible is a crucial MLOps principle to remember. The design, data processing, model training, deployment, and other machine learning documents should be well saved to guarantee that the model can be quickly reproduced with the same data input for MLOps to be successful in machine learning [12]. When MLOps are used effectively, each of the aforementioned principles supports the development of machine learning solutions that are ready for production.

E. Why Do We Need MLOps?

Whereas the stages and principles of MLOps sound fascinating and helpful for machine learning processes, a common concern from developers and enterprises is: we currently know about AIOps, why do we need to apply MLOps? To put it simply, despite their apparent similarity, MLOps and AIOps don't deal with the same issues. MLOps simply implements DevOps in machine learning initiatives. Let's take a look at some of the advantages and motivations for MLOps so we can better understand this [13].

F. Benefits and reasons for MLOps

1. Collaboration

In the typical setup, all teams working on an ML project, such as software development, data science, and IT operations teams, are seen as an uncorrelated and isolated group of people with diverse competencies. As a result of MLOps, machine learning models are built more collaboratively, with all teams working hand-in-hand to integrate their expertise.

2. Automation

Achieving targeted business outcomes necessitates the advantages of automating software and machine learning development By automating the ML-powered software's lifespan, diverse teams may focus on more important business concerns, resulting in faster and more dependable business solutions.

3. Quick Innovation

Because of MLOps' improved machine learning collaboration and automation, new, feature-rich machine learning solutions are being developed more quickly and efficiently [15].

4. Quick and Simple Deployment

Using a streamlined DevOps methodology, MLOps makes it simple to deploy a machine learning algorithm. It makes it easier to integrate, deploy, and deliver machine learning models continuously.

5. Effective Lifecycle Management

MLOps enhances the efficiency of all production teams as well as the complete machine learning project development workflow, from conception to deployment.

G. Machine Learning Pipelines

Every machine learning project involves teams of data scientists and machine learning engineers who work to develop cutting-edge models by hand or by computer. When selecting an ML model for training, data scientists consider how the model has changed over time, as well as the model's complexities [15,16]. Development/test environments differ from staging/production environments, and MLmanual process models frequently fail to adapt to changes in the dynamics of production environments or changes in data that represent production environments. Also. In the analysis of use cases, the machine learning (ML) manual pipeline technique is a convenient and widely used tool. The procedure is script-driven, and every step is performed by hand, even the most basic ones [16].

IV. FUTURE IN THE U.S

AI will expand the breadth and velocity of the DevOps movement even more in the United States of America's future. DevOps and AI may coexist peacefully if they are used in the right way. All industries' emphasis is now focused on end-to-end solutions that are smarter, faster, and more effective than ever before thanks to artificial intelligence. Using DevOps, firms are urged to increase the pace at which their technological solutions are delivered [17]. As DevOps strives to shorten the software development cycle, more changes may be expected in automation and event tracking. With more adaptability and innovation comes better security and dependability. In the context of DevOps, this makes perfect sense.

V. ECONOMIC BENEFITS IN THE UNITED STATES

There are several domains in which AI and DevOps may be beneficial to the US economy, including the customer value chain, supply chain, and backend procedures. This study will help the US achieve this information. Many businesses in the IT industry will benefit from AI and DevOps as they revolutionize business processes and enhance software development. AI-powered businesses will learn how to better their apps while dealing with low-quality code. Because of poor software quality, U.S. corporations are suffering, and only a small percentage of firms have used AI and DevOps in their operations [18]. This study will have a huge impact on enterprises like these, pushing them to use these technologies to cut down on the losses they incur while using them. These costs are often caused by delays resulting from subpar software and IT systems. Adopting AI-DevOps technology will boost the efficiency of many jobs, particularly in consumer services. To enhance their operations, companies such as IBM, Google, and Facebook will turn to the future of artificial intelligence and DevOps.

VI. CONCLUSION

This paper looked at reviewing Intelligent DevOps Platform Research and Design Based on Machine Learning. From this study, it's clear that the future of information technology will be centered on self-learning intelligent systems that use data to train on. This will be revolutionary, and businesses will seize control. Currently, businesses are working hard to improve the efficiency of their operations and production. Unlike typical software development, machine learning models have a different and more complicated lifetime, requiring substantial effort with extracting data, preparation and verification, infrastructure configuration and provisioning, post-production monitoring and updates, and other tasks. As a result, Agile principles/values, as well as DevOps approaches and technologies, are strongly recommended for providing continuous delivery and co-creation of value to the consumer, increasing the quality of the model, minimizing waste, and emphasizing the necessity of enabling a quick feedback loop, accommodating early adjustments, as well as exploring the underlying technological debt that leads to large increases in operating expenses for real-world machine learning algorithms. The actual success of DevOps in the IT industry will be shown via the use of deployment automation methodologies and an automated procedure for completing data analysis. Smoother delivery, fewer losses, and happier customers are all benefits of automation. This business model is slowly being adopted by a large number of enterprises. DevOps will be significantly aided by artificial intelligence (AI). The goal of DevOps is to concentrate computation and data under the supervision of AI. The use of DevOps procedures has simplified software testing and deployment. Advances facilitated by artificial intelligence will allow DevOps to be more productive.

REFERENCES

1) V. Gupta, P. Kapur and D. Kumar, "Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling", Information and Software Technology, vol. 92, pp. 75-91, 2017.
2) M. Ammar., "Application of Artificial Intelligence and Computer Vision Techniques to Signatory Recognition", Information Technology Journal, vol. 2, no. 1, pp. 44-51, 2002.
3) V. Sugumaran, Distributed artificial intelligence, agent technology and collaborative applications. Hershey, PA: Information Science Reference, 2009.
4) H. Salzman, "Engineering perspectives and technology design in the United States", AI & Society, vol. 5, no. 4, pp. 339-356, 1991.
5) H. Izadkhah, "Transforming Source Code to Mathematical Relations for Performance Evaluation", Annales Universitatis Mariae Curie-Sklodowska, sectio AI – Informatica, vol. 15, no. 2, p. 7, 2015.
6) J. Chen, "Discussion of the Modern Electronic Technology Application and Future Development Trend on Automobile", Applied Mechanics and Materials, vol. 155-156, pp. 627-631, 2012.
7) H. Papadopoulos, A. Andreou and M. Bramer, Artificial Intelligence Applications and Innovations. Berlin, Heidelberg: IFIP International Federation for Information Processing, 2010.
8) L. Rendell, "A new basis for state-space learning systems and a successful implementation", Artificial Intelligence, vol. 20, no. 4, pp. 369-392, 1983.
9) G. Pospelov, "Artificial Intelligence as a Basis for a New Information Technology", IFAC Proceedings Volumes, vol. 16, no. 20, pp. 1-14, 1983.
10) R. Conejo, M. Urretavizcaya and J. Prez-de-la-Cruz, Current topics in artificial intelligence. Berlin: Springer, 2004.
11) T. Bradley and T. Bradley, "Why DevOps means the end of the world as we know it", TechSpective, 2016. [Online]. Available: https://techspective.net/2015/08/16/why-devops-means-the-end-of-the-world-as-we-know-it/.
12) D. Linthicum, "What is DevOps? DevOps Explained | Microsoft Azure", Azure.microsoft.com, 2016. [Online]. Available: https://azure.microsoft.com/en-us/overview/what-is-devops/.
13) L. Iliadis, I. Maglogiannis and H. Papadopoulos, Artificial intelligence applications and innovations. Berlin: Springer, 2012.
14) Y. Jiang, "Analysis on the Application of Artificial Intelligence Technology in Modern Physical Education", Information Technology Journal, vol. 13, no. 3, pp. 477-484, 2014. [11] Y. Nakajima, M. Ptaszynski, H. Honma and F. Masui, "Automatic extraction of future references from news using morphosemantic patterns with application to future trend prediction", AI Matters, vol. 2, no. 4, pp. 13-15, 2016.
15) K. Hirasawa, "Trend on application of AI technologies to industry. From the latest international workshop on AI applications.", IEEJ Transactions on Industry Applications, vol. 108, no. 10, pp. 868-871, 1988.
16) L. Bass, I. Weber and L. Zhu, DevOps: A Software Architect's Perspective. Pearson Education,Inc., 2015.
17) L. Lopes, N. Lau, P. Mariano and L. Rocha, Progress in Artificial Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
18) G. Simov, "Artificial intelligence and intelligent systems: the implications", Information and Software Technology, vol. 32, no. 3, p. 229, 1990.