**International Journal of Computing and Digital Systems**
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

# The unison of CI/CD and Machine Learning

• **Anooshak Chowdhury[1] , Shreyas Mishra [1] ,Pooja Kamat[2] , Satish Kumar [3] ,and Madhan Raj Charles[2]**

[1] *Department of Information Technology, Symbiosis Institute of Technology, Pune, India*
[2]*Department of Artificial Intelligence and Machine Learning, Symbiosis Institute of Technology, Pune, India*
[3] *Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune 412115, India*

*Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##*

**Abstract:** The aim here is to have a discussion regarding the integration of Machine Learning applications and DevOps methodologies to make the latter more efficient. Even though machine learning is gaining popularity in almost every industry we can imagine, the process of making, delivering and updating the infrastructure crafted in ML is more complex to other software oriented applications, which is something the industry is taking maximum advantage of to facilitate frequent releases and updates for new features in a given application. Not to forget, a proper pipeline is a must in order to handle the workflow of constantly changing data that is variable in nature while simultaneously adding new and unique methodologies to the table to enable efficient working.If designed carelessly, models designed by ML engineers can lead to needless wastage of time that necessitates higher costs for managing the project. The paper here demonstrates how applying Continuous Integration (CI)/Continuous Deployment (CD) practices can help mitigate the problem mentioned in the former sentence. We know that with each and every day that passes by, the value of machine learning implementation in the real world gains traction.

**Keywords:** DevOps, Machine Learning, Azure, Pipeline, TDSP, AWS, Continuous Integration (CI), Continuous Deployment (CD)

## 1. Introduction

Within our current era, data science(with a specific facet of it, Machine Learning) is making waves in approaching solutions for the real world, evolving demands, and adding worth to their respective domains. As a result, a large group of professionals involved with Machine Learning and Data Science are looking for using descriptive and predictive models to improve overall business value. As a result, the ML Data Scientist and ML Operations Engineering teams are looking for ways in which DevOps principles can be applied to their field of work.

On the other hand, the cloud focused practices of Development and Operations, better known as DevOps, have gained attention for being a low cost way of creating and delivering software applications. Authors such as Kai et al. [5] proposed the need to incorporate practices relating to MLOps/DevOps along with IT based organizations. The aim here was to incentivize thorough understanding of the concepts of DevOps in-order to best take advantage of it in the coming future.

The practice of Continuous Integration (CI)/Continuous Deployment (CD) involves a development tactic of making updates to a given software based product come out with higher frequency. In this situation, the Continuous Integration  involves automatically making and testing to enable satisfactory merging of the code with the repository of choice, while the Continuous Deployment ensures smooth and timely delivery of the product.

The highly complex nature of CI/CD frameworks poses a challenge to fully integrate them in standard IT companies. To fully understand CI/CD, you need to understand them, test frameworks, software design, version control systems (VCS), and software delivery methods for all the various systems such as desktop, mobile, web servers, and cloud platforms. Another factor differentiating Frameworks are the features they provide, the platforms they are compatible with & the architecture alongside the language(s) they run on.

*E-mail:author's email*

## 2. Ease of Use

The following passages provide a rundown of DevOps (MLOps in this case) principles as part of a continuous Machine Learning Model deployment system in a given cloud platform.

### A. *Usability*

Usability, is simply put, an estimate or measure of how smoothly a specific goal can be attained by a given user at any given time. It is an important non functional requirement when dealing with any software based product, including Machine Learning & DevOps. This is one of the many takes on this non functional test.
Due to non-rigid nature of this function, usability has gotten many differing definitions through-out the years:

- "Usability is about human behavior. It recognizes that humans are lazy, get emotional, are not interested in putting a lot of effort into, say, getting a credit card and generally prefer things that are easy to do vs. those that are hard to do." – David McQuillen
- "The extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use." – ISO 9241-11
- "The ability for users to become familiar with and competent in a user interface (UI) in their first contact with that UI." - The Interaction Design Foundation

### B. *Development and Operations*

The above phrase is usually shortened to be referred to as DevOps[1]. This is a training pointed toward incorporating improvement, quality confirmation, and tasks (sending and mix) into one ceaseless arrangement of cycles. This technique is a characteristic expansion of Nimble and Constant Conveyance comes nearer from programming engineering standards. Be that as it may, DevOps is something other than a progression of activities. Rather, it is culture and reasoning that works with cross-useful group correspondence. One of the primary benefits of DevOps is that it plans to alter the manner in which groups work without necessitating significant specialized changes. Cooperation is an indispensable piece of the DevOps culture. The progress of the whole cycle depends on collaboration, upheld to various standards and practices that practitioners of this methodology follow.

#### 1. *Standards and Principles*

So, the fundamental standards of DevOps are computerization, constant conveyance, and quick response to criticism.

A more detailed description of the DevOps pillars can be found in the acronym CAMS [1].
Culture represented by human communication, technical processes, and tools
Process automation
KPI measurement
Feedback, best practices, and knowledge sharing
Continuous delivery, frequent Achieved a set of DevOps practices spanning deployment, QA automation, rapid validation of ideas, and team collaboration.

#### 2. *Models and Applications*

Devops necessitates a conveyance cycle that incorporates arranging, improvement, testing, sending, endorsement, and observing with dynamic coordinated effort among different colleagues, as displayed in Figure 1 [1].

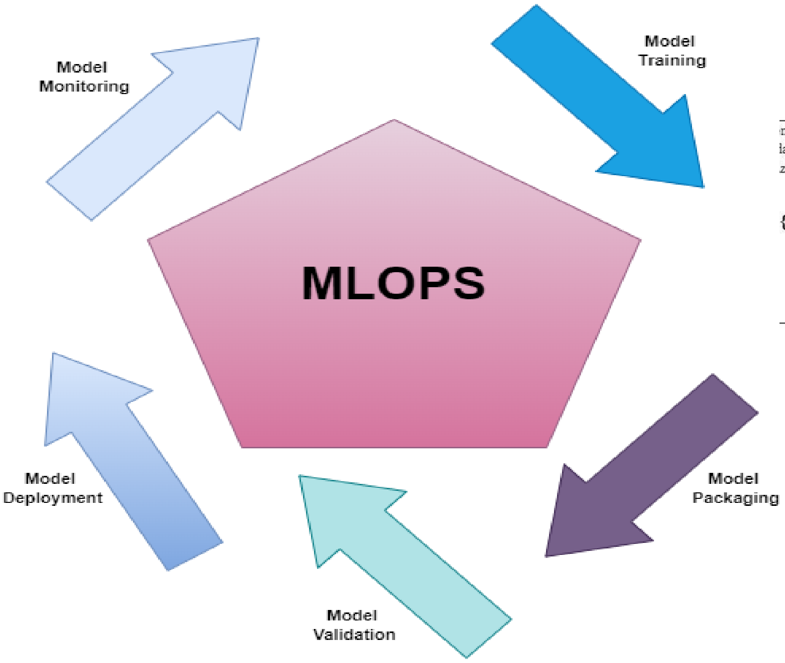


Figure 1. MLOps lifecycle

CD, an anagram for Continuous Delivery, is a methodology that incorporates advancement, testing, and deploy-ment tasks into one smoothed out process since it depends vigorously on robotization. During the improvement stage, engineers commit code in little lumps a few times each day for simple testing. Quality affirmation groups utilize mechanized instruments to characterize compulsory code tests. As mistakes and weaknesses are found, they will be delivered to the engineering department. The step additionally incorporates rendition control to expect mix issues. A form control framework (VCS) permits engineers to record document changes and offer them with other colleagues paying little mind to area. Code that passes computerized testing is united into a solitary common vault on the server. With successive code submissions, the distinctions between individual parts of code and the mainline code can become sensational over the long run, causing supposed "incorporation damnation" where combination necessitates more than genuine coding. can be forestalled. The most well known apparatuses involved in continuous joining happen to be Jenkins [3] and GitLab CI [4]. All source code is then conveyed and computed underway with open servers.

## 3. *Azure Architecture*

The architecture in this case can be handled for training custom models with ML methodologies. After this, our model can be deployed for further use. The working of the architecture can be viewed in the figure below:

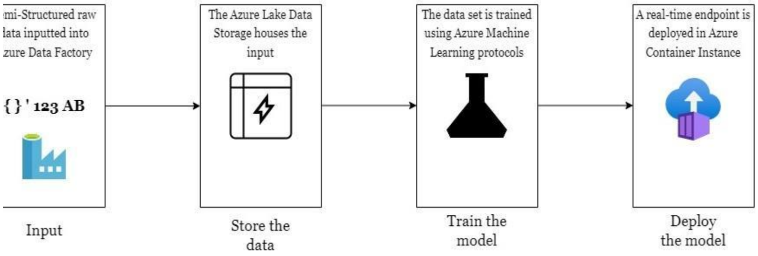

mi-Structured raw
lata inputted into
zure Data Factory
{ } ' 123 AB
Input
The Azure Lake Data Storage houses the input
Store the data
The data set is trained using Azure Machine Learning protocols
Train the model
A real-time endpoint is deployed in Azure Container Instance
Deploy the model

Figure 2. Azure Architecture

Figure 2 is a simplified layout of the architecture Azure supports for machine learning, with a brief explanation laid out below:

- Input – The data for usage (Eg. Json, csv, etc. ) is delivered to the Lake Storage. This data can be structured (geographic location), semi structured as displayed in the diagram (zip files) and unstructured data(audio samples).
- Storage – The inputted data is sent to the Lake Storage service.
- Model Handling – The service provided by Azure aids in making and sending models for deployment, which can be done even as an endpoint that is managed online.
- Deploy – The model created beforehand is then sent for deployment into the Container Instance.

## 4. *AWS Architecture*

In this service, a major focus is the automation aspect that has been stated to be aimed towards even folks who are not experienced with programming. the developers will be working mostly on the instance of sage-maker. Data pull-push is handled via the notebooks.

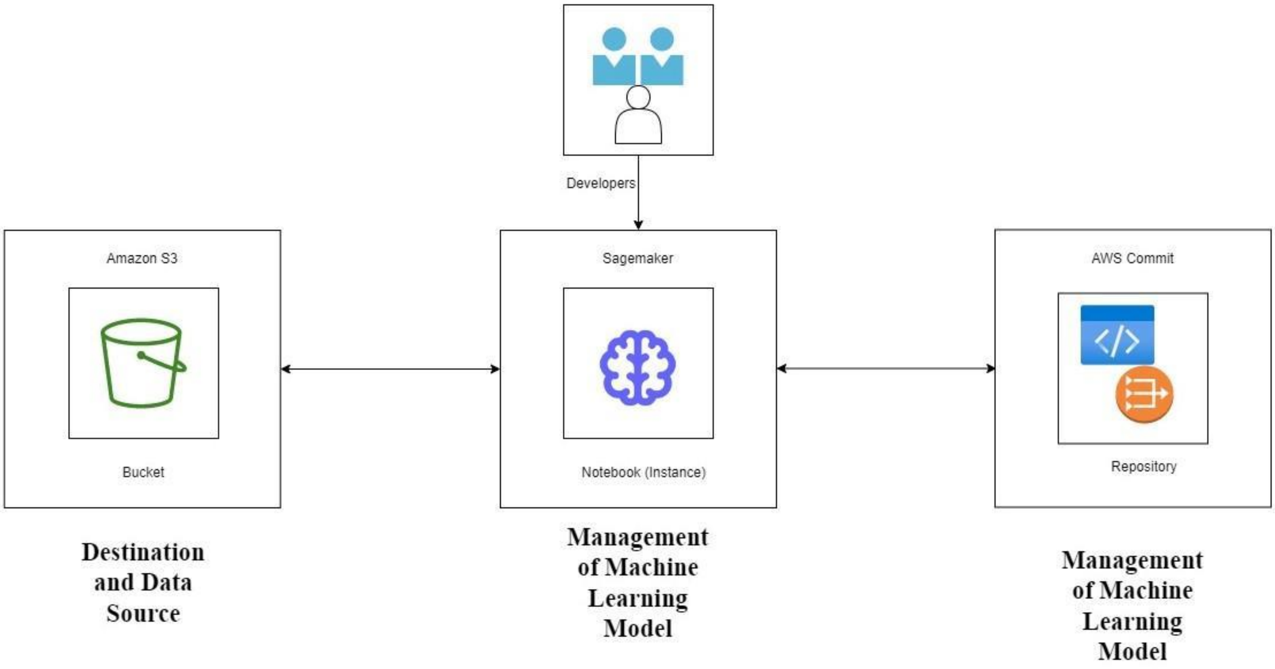
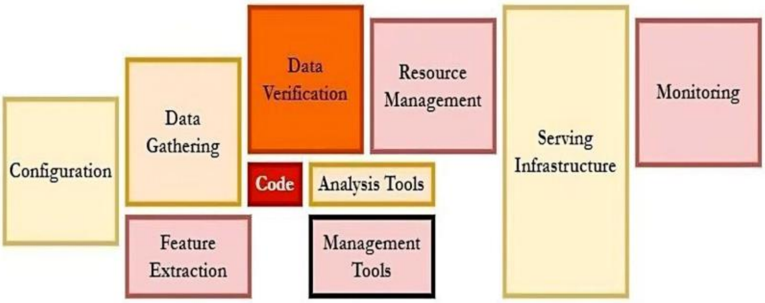

Developers
Amazon S3
Bucket
Sagemaker
Notebook (Instance)
AWS Commit
Repository
Destination and Data Source
Management of Machine Learning Model
Management of Machine Learning Model

Figure 3. AWS Architecture

A brief explanation of Figure 3. will be laid out below:

- Bucket (S3) – The data is as an object(s) within "buckets", where the object serves as the file (this includes meta-data serving to describe the stored file) while the bucket is a store for said objects
- Notebook – A computing instance under the EC2 server which runs in ipynb format used in python notebooks. This can be used to make and run a Jupyter notebook(s) for data handling.
- Repository – This enables us to store any code that has been made as separate documentations



## 3. METHODOLOGY

### A. Aim

The goal of this study has been to calculate the suitability a chosen CI/CD framework has for automation of ML model deployment. This study focused on a handful of MLOps concepts, and discussed 3 particular environments for deployment of models.

- Azure DevOps [2] allows designers to house source code, oversee item excesses, fabricate, deliver, and test programming, and store private programming antiquities.

- Amazon’s AWS Sage Maker [6] enables the developer(s) to personalize customer recommendations to improve business growth and automated detection of fraudulent activities.

### *B. Product Infrastructure*

The infrastructure of this project has a lot of important components to go over. Given down below is an overview of the cloud portion of the proposed system:

The Project Management Tool— Following a ML project is vital. It's not difficult to get lost and do a botch job while managing colossal information, highlights, ML code, assets and the executives. Fortunately, there are a great deal of ventures to take the executive devices out on the information highway to take care of us.

The Serving Infrastructure — When the model is created, tried, and all set, the next step will be to send it some place the clients can get to it. Most of the models are conveyed on a cloud platform of choice. Public cloud suppliers like AWS, GCP, and Sky blue even have explicit ML-related highlights for simple sending of models. Contingent upon the spending plan you can choose the supplier appropriate for your necessities.

Monitoring — It is of utmost priority to execute a checking framework to notice our conveyed model and the framework on which it runs. Gathering model logs, client access logs, and expectation logs will help in keeping up with the model. There are a few observing arrangements like graylog, elastic stack, and FluentD which can be used. Cloud suppliers as a rule transport their own checking frameworks. Figure 4 illustrates the structure of the application.

Figure 4: Architecture for the Sample Application

### *C. AWS vs Azure*

Fundamentally, the 2 services that we’re comparing here are aiming towards the same objective : ie. The act of automating the application of the created Machine Learning models into our daily lives alongside support of multiple learning methods (un/supervised), cloud based support or storing different features . That is where the similarities end.

- Firstly, the method of data preparation. While AWS Sagemaker pulls this off using it’s built-in features for processing, data wrangling, ground-truth labeling, Azure handles it through it’s unique synapse tool in tandem with the ingestion pipeline.
- The Sagemaker features native SQL support via Amazon’s Athena, whereas the Designer is required to handle such functions for Azure.
- Model quality observation is a major tool for monitoring the obtained data for AWS, while Azure relies on the logging of Payload to do this.
- License free installation of k8s via kubeflow is exclusive to the Sagemaker, while Microsoft’s equivalent has no such self-installation support.
- Permissions for using a workspace(s) is available by default in Azure, while AWS needs the user to manually enable the required permissions.

### *D. Comprehension of the Profession*

In this phase of the Group Information Science Cycle, there are two principal assignments to concentrate on

Characterizing objects Information researchers and visitors, as well as different partners, work togeth-er to distinguish and figure out issues in the involved business, think about how to ask what defines the business assumptions. The go-to involved with the step deals with naming significant factors in the business, which is then put through scrutiny . These elements are considered to be the ' imitable targets ', rules of which are used to choose the advancement of the arrangement. Define the business suspicions by presenting and further developing requests that are material, express, and unequivocal. Describe the unit by showing the spots and liabilities and produce an elevated place corner plan that emphasizes as additional data is dis-shrouded. Characterize achievement standards as Brilliant( explicit, quantifiable, feasible, relevant, time-bound).

Distinguishing wellsprings of the information identify the material information that provides assistance to break the inquiries that characterize the objects of the plan. Distinguish information sources that contain the responses to questions and ensure that the information is pertinent to the issue. Additionally, this information should have a precise proportion of the objective model and the highlights that count.

### *E. Collection and Comprehension of Data*

This basic stage centers around truth risking the information. It begins by ingesting the da-ta, investigating and setting up an information channel. The assumptions of this stage are to create a perfect, great dataset that can be connected with the objective factors. Find the informational collection in the appropriate examination landscape to begin the demonstrating system. produce a solution armature of the information channel that refreshes the information routinely.

Assimilate the information, arrange the channel to shift information from any given source regions to the destination districts, followed by investigation activities.

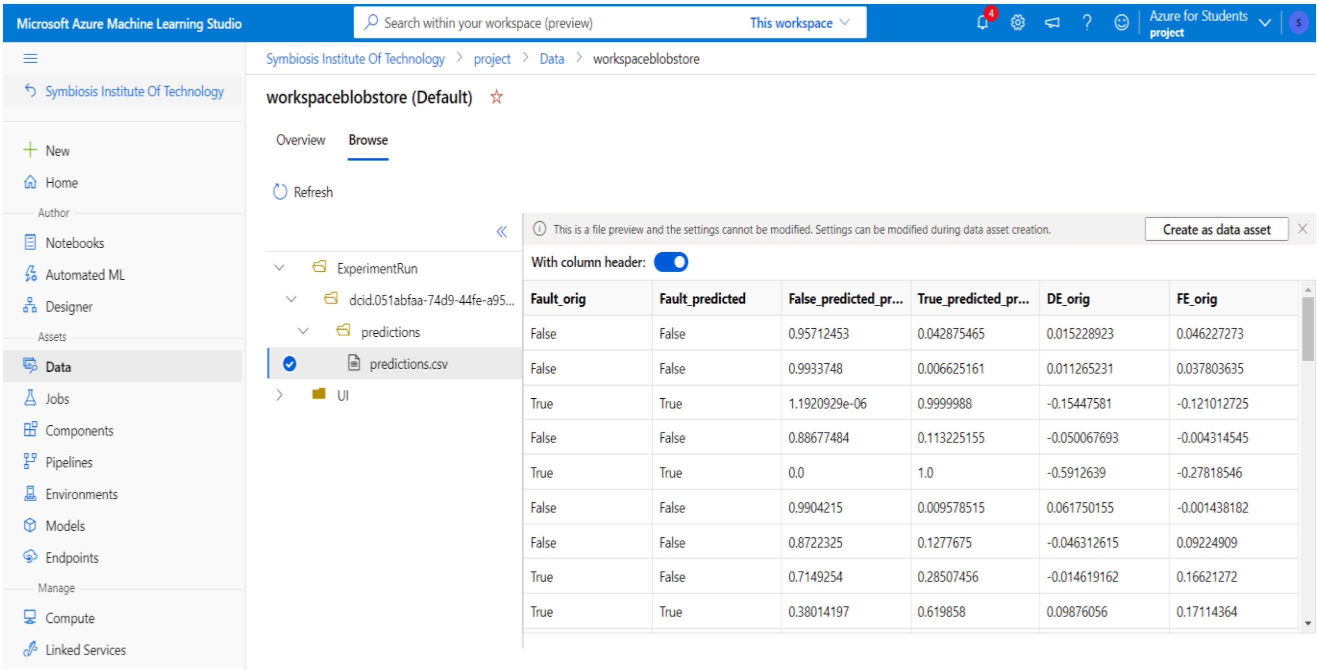
Investigate any information obtained for preparing the model, in order to result in fostering comprehension of the information. This present reality dataset is habitually loud, with various unavailable qualities and/or outliers. Information representation may be utilized for auditing the information nature; moreover, the necessary data to set up the information before it's prepared for demonstrating gets delivered.

Arrangement information notwithstanding the first ingestion and blessing of the information, by and large bear setting up an interaction to modernize new information or invigorate the information regularly as a feature of a relentless education

## 4. RESULTS

Utmost — or, for pall- grounded fabrics, all - of crafted by upholding these line lines included arranging their activity. Regular contemplations incorporate indicating job type, Adjusting any channel for the desired objective working framework, relating line factors requested to design an edge and semantics. Predefined strategies are the types of tasks that suddenly spike in demand for an edge's representative. Errands can bear different send off methods on various oper-ating frameworks or neglect to harborage by and large e.g., commands from PowerShell not running properly or at all in a Linux specialist. Pipeline factors contrast as well. In Order to properly design a channel, the given framework explicit contrasts should be connected, dove , and tended to. CD, an anagram for Continuous Delivery, is a methodology that incorporates advancement, testing, and deploy-ment tasks into one smoothed out process since it depends vigorously on robotization. During the improvement stage, engineers commit CD, an anagram for Continuous Delivery, is a methodology that incorporates advancement, testing, and deploy-ment tasks into one smoothed out process since it depends vigorously on robotization.

Here, a sample python notebook with basic machine learning protocols implemented on the IR-7 and NB Normal datasets has been deployed. DE (The Drive End of the sensor) and FE (The Fan End of the sensor) serve as the input values, with Fault as the output. Using the aforementioned tools provided by Azure, not only have the Fault been computed , but also computed the true and false predictions, shown down below in Figure 5.



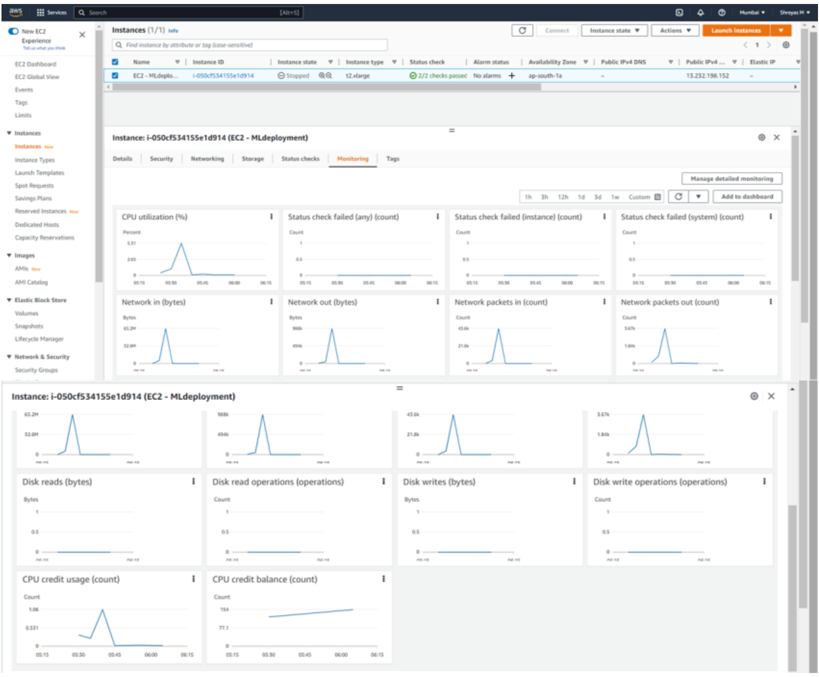
This is a file preview and the settings cannot be modified. Settings can be modified during data asset creation.

With column header:

| Fault_orig | Fault_predicted | False_predicted_pr... | True_predicted_pr... | DE_orig | FE_orig |
|---|---|---|---|---|---|
| False | False | 0.95712453 | 0.042875465 | 0.015228923 | 0.046227273 |
| False | False | 0.9933748 | 0.006625161 | 0.011265231 | 0.037803635 |
| True | True | 1.1920929e-06 | 0.9999988 | -0.15447581 | -0.121012725 |
| False | False | 0.88677484 | 0.113225155 | -0.050067693 | -0.004314545 |
| True | True | 0.0 | 1.0 | -0.5912639 | -0.27818546 |
| False | False | 0.9904215 | 0.009578515 | 0.061750155 | -0.001438182 |
| False | False | 0.8722325 | 0.1277675 | -0.046312615 | 0.09224909 |
| True | False | 0.7149254 | 0.28507456 | -0.014619162 | 0.16621272 |
| True | True | 0.38014197 | 0.619858 | 0.09876056 | 0.17114364 |

Figure 5. Implementation in Azure

In the AWS implementation, a handful of graphs can be viewed in Instance : i-050cf534155e1d914 in the figure below.



| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4 ... | Elastic IP |
|---|---|---|---|---|---|---|---|---|---|
| EC2 - MLdeplo... | i-050cf534155e1d914 | Stopped | t2.xlarge | 2/2 checks passed | No alarms | ap-south-1a | – | 13.232.198.152 | – |

Instance: i-050cf534155e1d914 (EC2 - MLdeployment)

Figure 6. Implementation in AWS

Here, in Figure 6, through the monitor section of the deployed model, it can be observed that the statistics of the deployment, such as CPU credit balance (the currency for operating aws ec2 on a given device), network consumption (network in and network out displayed in the middle left) and the network packets at the middle right.

4.1.1 AWS Cloud. The AWS Cloud Arrangement project included making a Flexible AWS Beanstalk housing various cases (Docker Holders). Every example was designed with a straightforward .NET Center 2.2 MVC Layout Application in mind. Azure DevOps was chosen to give a CI/CD pipeline involved with the project, while EC2 gives the server to send in.

4.1.2 Azure DevOps. It is an across the board structure for overseeing programming advancement undertakings, including the executions of CI and CD. It gives 2 principal setups: Assemblies and Deliveries. The Assembly(ies) Pipeline gets usage during arranging, bundling software for sending. The Delivery(ies) Pipeline gets usage with de-endorsed for arranging Fabricates bundled curios.

## 5. CONCLUSIONS

The white paper presented a mechanized ML pipeline utilizing CI/Cd standards. This strategy depends on DevOps rehearses liable for coordinating and sending models prepared and tried in ML. Different ML techniques have been introduced, however the TSDP strategy is more reasonable for this review. Manual ML strategies have been found to cause high functional expenses and deferrals for business associations. The proposed computerized ML procedure further develops numerous regions, for example, time-to-advertise, mix between specialty units, and end of storehouses between divisions. It additionally further develops code and organization quality, efficiency, and perceivability. Be that as it may, it isn't clear; many organizations battle and stick toward the beginning of the excursion while others cut short the execution partially through the cycle because of moves like protection from change, separated groups (storehouses), absence of ranges of abilities, and so forth. It is basic to understand that DevOps doesn't remain solitary, yet it depends on the reception and joining of various structures and strategies like ITSM and Light-footed.

AI model lifecycles are unique and mind boggling compared to the standard programming improvement; it requires broad work with information extraction, prepara-tion and confirmation, foundation setup, provisioning, after creation checking, and upgrades. In this way, Lithe standards/values and DevOps prac-tices and devices are strongly prescribed to give persistent conveyance and co-making of significant worth to clients, increment the model quality, limit squander, high-light the significance of supporting a fast criticism circle, oblige early changes, as well as investigate the secret specialized obligation that prompts a tremendous expansion in operating expenses of genuine AI frameworks. The proposed Machine Learning pipeline strategy includes DevOps standards and practices; hence, the benefits mentioned above are added esteem in theML model(s) lifecycle.

As far as the platform comparison goes, each of the 2 platforms have their unique advantages and disadvantages; with AWS being effective with simpler, small scale models with ease to setup. While Azure is usually more flexible with containment support and flexible UI, which ultimately led us to choose Azure for our own college project.

By utilizing our strategy, future work could zero in on unambiguous parts of models of complicated frameworks and foster helpful programming to help testing of ML models [7].

## 6. REFERENCES

[1] DevOps Documentation. Available online: https://devops.com

[2] Microsoft. 2020. Azure DevOps Services. Microsoft. https://azure.microsoft.com/ en-us/services/devops/.

[3] Chacon, S.; Straub, B. Pro Git, 2nd ed.; Apress: Berkeley, CA, USA, 2014.

[4] Christof, E.; Gallardo, G.; Hernantes, J.; Serrano, N. DevOps. IEEE Softw. 2016, 33, 94–100.

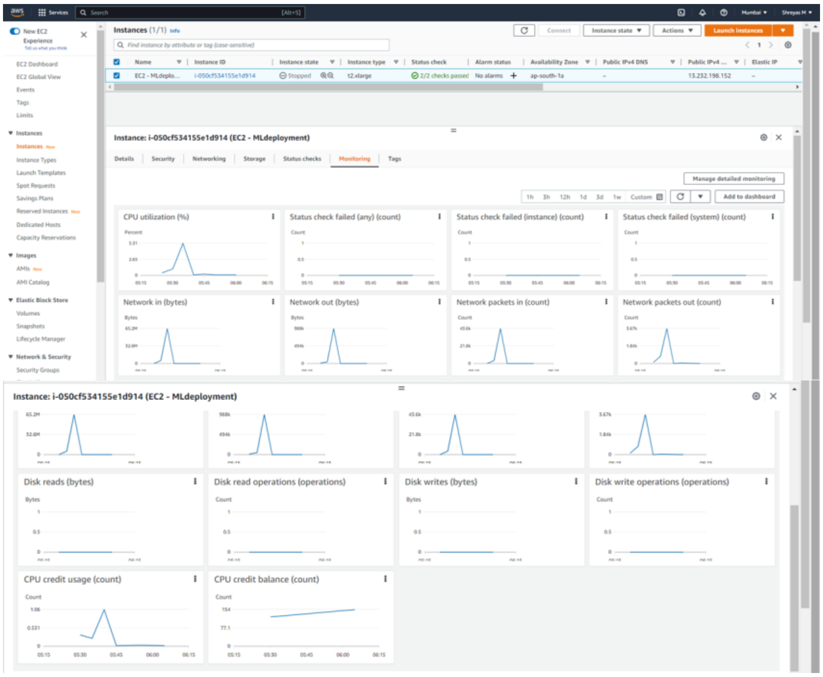
[5] Ramtin Jabbari, Nauman bin Ali, Kai Petersen, Binish Tanveer. 2016. What is DevOps?: A Systematic Mapping Study on Definitions and Practices. In Proceedings of the Scientific Workshop Proceedings of XP2016. What is DevOps?: A Systematic Mapping Study on Definitions and Practices

[6] Amazon SageMaker. 2017. AWS Services. Amazon. Amazon SageMaker

[7] Microsoft Azure v/s Amazon AWS Cloud Services: A Comparative Study , Microsoft Azure v/s Amazon AWS Cloud Services: A Comparative Study

**Anooshak Chowdhury** is a student currently pursuing Btech in Information Technology(IT) at Symbiosis Institute of Technology, Symbiosis International (Deemed university), Pune, India.His current interests lie in Data Science and Business Analysis, being backed by competence in various technologies as Python, C/++, Excel, Ubuntu, Power BI, etc. His research interests lie in the fields of MLOps, Bayesian Functions, Predictive Analytics, Economical Data Science, etc.

**Shreyas Mishra** is a student currently pursuing Btech in Information technology at Symbiosis Institute of Technology, Symbiosis International (Deemed university), Pune, India., . Cloud & web development enthusiasts mainly interested in devops field creating handful applications with pipeline, having hands on experience in python,C++, SQL, Linux - and framework such as flask & pyspark.His Research interests include version control management,CI CD CM of applications,operation on test cases with pipeline,automation tools linkable to multibranch pipeline and devops operation with dynamic APIs and deployment of those APIs



Instances (1/1) Info

| ☑ | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4 ... | Elastic IP |
|---|---|---|---|---|---|---|---|---|---|---|
| ☑ | EC2 - MLdeplo... | i-050cf534155e1d914 | Stopped | t2.xlarge | 2/2 checks passed | No alarms | ap-south-1a | – | 13.232.198.152 | – |

Instance: i-050cf534155e1d914 (EC2 - MLdeployment)

**Prof. Pooja Kamat** works as an Assistant Professor in the Computer Science Engineering and AI & Machine Learning department at Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, India. She has completed her M.Tech from Mumbai University and is currently pursuing Ph.D. in the domain of Predictive Maintenance. She has a teaching experience of 12 years and has guided many UG and PG students in the domain of Artificial Intelligence and Machine Learning. Her research interests include Predictive Analytics and its application across various domains. She has authored many international/national journal and conferences publications.

According to Google Scholar, she has 300 + citations, with an H-index of 9 and an i10- index of 9.

**Dr. Satish Kumar** is an Associate Professor in the Department of Robotics and Automation at Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India. He did his Master's degree (M.Tech) and Doctoral degree (Ph.D.) in 2013 and 2020 from K Visvesvaraya Technological University, Belgaum, Karnataka, India. He has 8+ years of experience in teaching, research, and industries. His research interests include Smart Manufacturing, Condition Monitoring, Composites, Cryogenic Treatment, Additive manufacturing, and Hard materials machining. He has authored more than 30 international/national journal and conference publications leading to h-index. According to Google Scholar, he has 250 + citations, with an H-index

**Madhan Raj Charles** is a student currently pursuing Mtech in Artificial Intelligence and Machine learning (AIML) at Symbiosis Institute of Technology, Symbiosis International (Deemed university), Pune, India., alongwith He is a Research Intern at Symbiosis Centre for Applied Artificial Intelligence (SCAAI) , Symbiosis International. He holds a bachelor degree in Electrical & Electronics Engineering (EEE) from FET, Jain(Deemed-to-be-university). His Research interests include use cases in Explainable AI , Data driven Digital twin , Computer vision, Advanced AI Defence-tech.