IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

ISO 3297:2007 Certified ⫶ Impact Factor 7.12 ⫶ Vol. 10, Issue 12, December 2022

DOI: 10.17148/IJIREEICE.2022.101216

MLOps at Scale: Bridging Cloud Infrastructure and AI Lifecycle Management

Phanish Lakkarasu

Staff Data Engineer, ORCID ID: 0009-0003-6095-7840

Abstract: To successfully manage the development and deployment of machine learning models (ML models), organizations require a platform that provides the necessary tools, standardized settings, and workflows for ML teams to easily build, monitor, and leverage ML models' intelligence and insights at scale. This introduction section highlights the challenges organizations face in building robust, automated, reliable, and production-ready machine learning solutions. These challenges include managing changes across multiple tools, monitoring components throughout the AI lifecycle, and enabling collaboration among all teams involved in this lifecycle. Organizations need a hybrid cloud infrastructure that allows them to connect MLOps tools hosted on different clouds and vendors while enabling low-maintenance integration and a simplified, extensible developer experience. Traditionally, data scientists and AI developers performed data and model storage, training, and predictions on the organizations' infrastructure, whether on-premise or cloud-hosted. These systems allowed the development of ML solutions at scale through clustering hardware or distributed training techniques. These solutions were typically written as self-contained batch programs scheduled and monitored with existing data processing schedulers. The majority of ML workloads were still too simple for the engineering and deployment work needed to develop a robust and efficient solution, which allowed for other approaches to be followed. Once models were defined and trained, they were typically saved to disk, logged into a version control system, or manually documented in the source code or ticketing system. Existing systems, which in general were made for more traditional software, did not efficiently address the needs of an ML development team, which had very different tooling and priorities. This introduction focuses on vendor-neutral and cloud-agnostic approaches to the MLOps platform that empowers organizations to choose or easily integrate multiple open-source or proprietary tools into their workflows and pipelines while abstracting them with a streamlined API. The proposed platform addresses the aforementioned challenges faced by organizations by offering a set of deployment-ready components, giving them more freedom for customizing their MLOps and AI infrastructure management. Finally, the achievements of the MLOps works mentioned above and expected contributions to the literature are discussed.

Keywords: Data Science Platform; Data Lifecycle Management; Deployment & Monitoring; Machine Learning Platforms; MLOps. MLOps, MLOps system; ML; Ai; AIops; AIOps; Development process.

I. INTRODUCTION

Artificial intelligence is becoming a crucial part of businesses looking to capitalize on the benefits of AI and ML models. AI systems, like humans, may sometimes behave in an unwanted and complex manner. MLOps versions are presented, along with their benefits, difficulties, evolutions, and important underlying technologies such as MLOps frameworks. MLOps is a pipeline of workflows divided into four steps: model design, model training and tuning, deployment, and operations. The MLOps framework tools are explained with classifications into model exploration and deployment and data exploration and deployment. An end-to-end production of ML projects is presented, along with the maturity levels of the automated pipelines, which vary from no automation to complete CI/CD and CT capabilities. A detailed example of an enterprise-level MLOps project for an object detection service is used to explain the workflow of the technology in the real world.

Over the last two decades, there has been a lot of progress in the field of Artificial Intelligence (AI), especially Machine Learning (ML). Many variables come into play, from prediction variables to internal aspects, making it very difficult to understand why such a result occurred in a classy or unexpected manner. This black box aspect of AI systems makes it hard for engineers and managers to understand what is happening inside the model. In addition, the complexity of the models and the vastness of the information space require a team effort to create solid ML systems. The need arises for ML practitioners to create this environment, which should be accessible and collaborative, reproducible for mass production, and monitoring over time. It is necessary to define both restrictions and stages for the model simultaneously using Agility and best software engineering practices. The use of Machine Learning Operations (MLOps) shall be necessary to create this environment.

The MLOps positions in the companies of various sizes and their roles and responsibilities are explained. It is frequently depicted as an extension to DevOps or DataOps. The goal is to facilitate and automate the collaboration, development, and operations of end-to-end ML workflows, one of their various definitions. Those workflows are not easy to describe, as they differ highly on each company, which is why this initial definition can be very generic. Also, this is true for different DevOps positions, which vary at least as much in different firms.

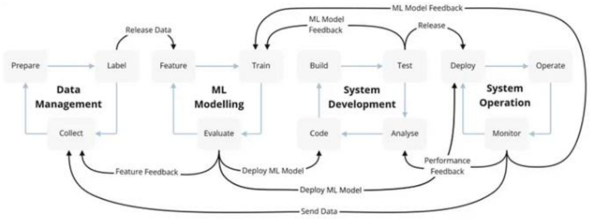

Fig 1: Introduction to MLOps

1.1. Background And Significance

With the rapidly growing capacity of cloud and on-prem machines, today's machine learning (ML) solutions have unprecedented potential for analyzing and forecasting data. Massive amounts of data can now be processed by increasingly powerful models for complex tasks, such as image analysis, text processing, and code generation. This is widely thought to be Artificial Intelligence (AI). AI has the potential to enhance a company's effectiveness (i.e., increase income, cut expenses, etc.). However, developing machine learning models and AI products entails numerous stages, often ranging from thirty to a hundred. It is just as important for the models to function reliably and accurately in the production environment as any other part of the software stack. Therefore, ML providers must deliver and maintain their models, just as application service providers must with their applications. This model delivery system with attention to quality is typically referred to as ModelOps (MLOps). More specifically, the cloud-scale MLOps solutions that the cloud service providers offer.

MLOps addresses the entire life cycle of the ML model, which goes beyond cloud infrastructure. Using any of the popular cloud service providers (CSPs) and their integrated MLOps systems, the need to manage the infrastructure, such as provisioning VMs, configuring hard disk size, and adjusting networking security, is done with infrastructure-as-code practices. Also, MLOps has a python SDK that creates the datasets and preprocesses them in the Pandas dataframe style. The hyperparameter optimization is a straightforward job after training the model pipeline. Several UI interfaces allow non-tech-savvy consumers to do modeling with fewer script codes. On top of that, the voice and eyesight ML platforms allow users to drag and drop audio data and images to build more baseline models without coding. This evidence supports the evolution of MLOps at the brink of suitability.

Nevertheless, all these prior infrastructures and suggested ways in MLOps are about handling a part of it. There is much research on cloud infrastructure design and reasoning. However, little focus on the integration of AI lifecycle management and cloud infrastructure units is put. It is a missing piece because it gives power to leverage the abundant resources of the cloud and design a systematic AI platform unlike before that bridges the ML development and production processes. MLOps at Scale, addressing this issue, are nouns that take it as a system.

For many years, successful businesses have been exerting persistent effort and investment trying to improve the value of their products and services while finding new ways for differentiation. One commonly accepted way of doing this is through the careful application of data science and machine learning (ML). Over the last few years, the interest in Artificial Intelligence (AI) and Machine Learning (ML), which is a sub-area of AI providing better decisions based on learned experiences, has rapidly grown. Many successful companies have made efforts to competitively utilize the so-called Data-Driven Decision-Making (DDDM) paradigm based on data obtained from customer interactions with the companies, from business operations, etc. Big Internet and Business Intelligence companies are allocating substantial resources in attracting data, not only as a means for doing business but as a potential resource for completely new ones. However, studies showed that while many companies are investing heavily in data collection and modeling, they report only weak results in the area of successful ML implementation in their routine activities.

Machine Learning Operations (MLOps) is rapidly becoming an important part of the enterprises trying to benefit from the AI and ML models. The growing industrial adoption of MLOps has brought a renewed interest in understanding and advancing the field, and many researchers are contributing to this domain from various perspectives: MLOps frameworks, tools, pipelines, jobs using AI and ML architectures, including edge and cloud, migrations of infrastructure, and AI supply chain including data sourcing and supervision. focused on presenting a high-level understanding of MLOps, discussing not only benefits and difficulties in the framework of a research organization but, unexpectedly, also showing that, in industry, the process is much more streamlined than it is in academia; thus, there is no need for a research spin-off at the current moment. Many companies offer a huge variety of high-level cloud services covering end-to-end AI supply chains from data deals to AI architectures. Some of them allow end-users to create their own tools, opening a door for many research opportunities. At the same time, most of the popular independent MLOPs tools are being quickly absorbed by one or the other cloud provider. In the area of operating with knowledge (explainable AI), a lot of effort is needed in connecting inconsistent data sources and checking the value of data.

Equ : 1 Model Deployment Velocity

$$V_d = \frac{N_m}{T_d + T_v}$$

- V_d = Deployment velocity (models deployed per unit time)
- N_m = Number of models ready for deployment
- T_d = Average deployment time per model
- T_v = Average validation time per model

II. UNDERSTANDING MLOPS

With the explosive growth in AI models and platforms, organizations are being pressed to develop MLOps (machine learning operations) systems to manage the ML lifecycle and ML value chain holistically and responsibly across their enterprise. The emergence of Large Language Models (LLMs) and Foundational Models has further fueled this interest. MLOps systems enable organizations to deploy reliable and effective ML solutions to generate business value, drive efficiency, and monitor compliance with trustworthy AI objectives across different business verticals such as finance, healthcare, retail, and telecommunication.

MLOps serves the intersection between cloud infrastructure and ML lifecycle management of broad generative and domain-specific AI tools. This section provides an understanding of MLOps through a narrative supported by broad references to seminal works and current practice. MLOps, in several senses, refers to both the systems and practices for managing the ML lifecycle and developing and producing ML models. MLOps systems should be capable of working as a collective, constant, repeatable, validated, and monitored in a way to meet corporate MLOps objectives. Corporate MLOps objectives can be values for state variables, the function to be optimized, or constraints to be satisfied [3]. Broadly speaking, there are three components of MLOps development process: (a) model: objects such as data, terminology, and business context are represented and used to develop ML models; (b) code: MLOps systems must operate master pieces of code, typically in notebooks, and other sources, documentation, and artifacts related to the ML models; (c) data: MLOps systems should operate data in exhaustive manner and start their operation pipelines as automatic data constraints are broken. Most MLOps systems exhibit primarily or exclusively observability, monitoring, tracking, and reporting of what is going on in relation to MLOps development process. In addition, MLOps systems can periodically and automatically execute all and several of the monitoring aspects of the MLOps development process such that models are analyzed (e.g., signals like drift, outliers, leakage are reported) and maintained. These assertions apply under the scrutiny of a scientific eye and model misbehavior in relation to thresholded conditions, regression tables, etc. In this trigger when something does not behave as expected data is ingested, retrained with exploratory analysis, and a transferable artifact is committed to the model registry.

Recently, many organizations have reported challenges turning an ML proof-of-concept into a production-quality AI system. The experimental nature of the ML development limits many qualities: reproducibility (the ability of a user to replicate specific ML behavior), the ability to perform hypothesis testing on data and code modifications, testability (the ability to run tests on the AI system and the hyper-parameters used), the ability for a receiving end to exponentiate trust in it (no black-box systems), and traceability (high transparency). These qualities are needed when putting a trustworthy product or service on the market, from which profitability becomes possible. MLOps can be seen as a mindset on the highest level. However, this is not a purely abstract matter. MLOps can also be seen as an engineering discipline. Organizations adopting MLOps hope to deploy and maintain ML systems in production reliably and efficiently. This involves embracing a culture with corresponding processes that an organization must adapt for the specific application domain, something that takes time, preferably in steps.

DOI: 10.17148/IJIREEICE.2022.101216

Turning to specific elements of MLOps, automated ML requires that development, validation, and deployment can happen either fully automatically or using GUIs that do not require ML or programming expertise. MLOps relies mostly on pipeline automation to remove the barriers between data processing, model training, model testing, and model deployment. Some pipelines become a fundamental component in a MLOps environment: The connected pipeline performs the ML development work from raw unprocessed data to an estimation of prediction performance. The interfered one allows for all modifications of any step in the connected pipeline (including outside it) to be monitored, and the effects on the performance of the rest of the pipeline investigated before accepting or rejecting the modifications in the connected pipeline. Experimentation with and evaluation of candidate ML components are handled as first-class citizens in a MLOps environment, with facilities such as testing at scale and retraining strategies.

If not designing for the operations phase, it will be tough to reach sustainably value-creating AI solutions. This does not mean that the parallel organizational competencies and processes needed to build an MLOps set-up will not need to be educated. MLOps can be thought about in general terms, but the implementation must be more technical. MLOps is not yet well sought after from an academic perspective, that MLOps phenomena would be better studied in the field than in extensive and heavy university labs is an idea worth pursuing. These MLOps solutions are the focus of this piece. They are currently somewhat piloted, but the hope is that they will become reliable and reusable when concrete AI development projects with societal potential start up.

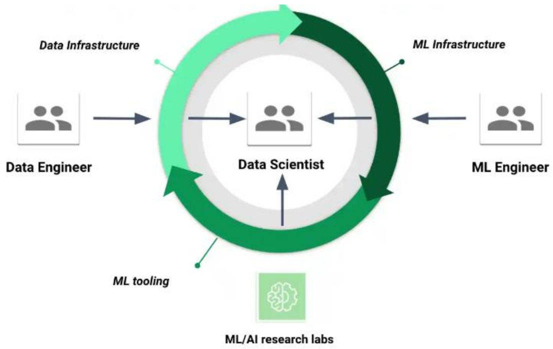

Fig 2: Understanding MLOps

2.1. Definition and Importance

MLOps is a framework that unifies the demands of both data scientists and a company's cloud infrastructure/security engineer organizations as AI/ML practitioners' workloads migrate from experiment for proof-of-concept purposes to full production. Jobs evolve first to smaller groups requiring limited cloud infrastructure management knowledge for development/experimentation and proper security controls. The effort is focused on instance size, costs, and supply chain provenance of products deployed in public clouds. Secrets are dynamically injected into environments post-containerization, potentially via cloud infrastructure-as-code implementations, and in containers. The data scientist is supported by the development of high-level notebooks and user interfaces that allow the necessary functionality without in-depth knowledge of containers, command line or programming languages, and cloud infrastructure security practices. MLOps at scale should help recognize the identifiable AI/ML workloads that need such effort for transition to secure cloud production. The identification relies on previous AI/ML workloads and scaling metrics like the number of users and models. Governance comes from a matrix of cloud infrastructure responsibilities and AI/ML practitioners' responsibilities, possibly relying on existing security organizations.

AI and ML pipelines are defined and instantiated from low to high end-user/no-code environments to infrastructure developers engaging low-code containerized environments. With all notebooks saved in version controls, cache capabilities, and consistent run-time management, the ease of use of previous shared notebooks can be increased. MLOps for tenants should provide visualization, analytics, and cost estimate features to help define the sandbox. Data engineers are supported with ingestion template transform workflows across sources and targets, while fortifying governance and compliance. In situations of extensive data, pipelines provide random small target samples as the data science workbench visualizes and investigates the samples. The risk of loss and potential abuse is minimized through collaboration with security engineers to classify and mask workflow data.

Machine learning (ML) and artificial intelligence (AI) are becoming increasingly important and prevalent in modern societies, enabling improvements in numerous aspects of personal and professional endeavors. In many companies, departments, and institutions, self-contained, proof-of-concept, or pilot ML systems may be commonplace.

DOI: 10.17148/IJIREEICE.2022.101216

Still, it is relatively rare to find ML systems or related AI systems that can be deemed operational. Converting a prototype into a production-quality AI system is challenging due to the experimental nature of ML, which limits qualities such as reproducibility, testability, traceability, and explainability. MLOps (ML/AI) is a set of practices that combines ML, DevOps, and Data Engineering. It involves embracing a culture that corresponds to the processes that an organization must adapt for the specific application domain.

Without designing for the operations phase and ensuring push-button changes of ML models to production, it will be tough to reach sustainably value-creating AI solutions. Recently, many organizations have started experimenting with generic MLOps sectors. Many challenges need to be addressed, as most MLOps solutions suffer from various shortcomings. In many organizations, the current solutions related to creating operational ML systems are perceived as deficient. Typical data scientists spend ten (10) hours a week on ML model and data management issues. Some of these organizations have opted for a wholesale switch to one specific cloud vendor and its version of MLOps. But these travels often face backlash from individual practitioners.

MLOps allows for pipeline automation to remove the barriers between data processing, model training, model testing, and model deployment. At the same time, it enables deploying and maintaining ML systems in production reliably and efficiently. It is an approach that numerous companies and startups are pursuing today, both in terms of individual products and packaged services; it is not yet well-researched from an academic perspective. The general principles surrounding MLOps are already known and sometimes even implemented in teams and development practices. However, creating operational systems has proven challenging, especially in larger teams or organizations.

2.2. Key Components of MLOps

Machine Learning Operations (MLOps) is a crucial component in scaling Machine Learning (ML) applications that ensures a well known versioning methodology, high observability for serving applications, a robust development pipeline relying on Continuous Integration/ Continuous Deployment (CI/CD), among others. Typically, the bundle of servers that enables scalable infrastructure and DaaS companies is called DataOps. It is now critical that the link from MLOps to DataOps, i.e.MapReduce(-like) architecture is developed, and scalability is guaranteed. In a recent enterprise-wide survey of ML systems deployed in large organizations, responses selected and implemented solutions across Cloud-based Infrastructure as a service (IaaS) or Platform as a Service (PaaS), workflow automation for continuous data and model pipelines, cloud schedulers for orchestration and deployment management, container-based deployment, as well as testing and monitoring systems for observability of services.

This review illustrates the bottlenecks in enterprise scale MLOps based on Niagara network research experience, and presents a working system of MLOps at scale on Google Cloud infrastructure with various tools, including but not limited to definitions of components across Google Data Analytics and ML Cloud Solutions, operated pipeline and AutoML, Open Source tools and Data analytics pipelines across. Associated high-level architecture, enterprise-scale workflows, user stories, monitoring dashboards, and performance testing results for various generic properties are also shared for transparent understanding of end-to-end service development topology and quality control. Their design principles, extension, and benchmarking recommendations will be discussed. An open discussion will finish with interesting queries and perspectives for MLOps and related disciplines.

Two categories can be used to represent the complications involved in automating AI workload and reporting observability. The availability and relationship of tools provided by a common cloud provider with data big-data engineering and hyper-parameter tuning directly affects AI team productivity. A budget-decided-migration-strategy-favoured-containerized-accelerated-cloud-implementation-on-pre-existing-on-premise-bio-informatics-pipeline over-hunted productive AI, big-data, and DevOps-tools instances. On public cloud, auto-scaling distributed-worker-resources and 24-hour-availability of used-able-standard-training-configs, efficient training time required and widely available adapted pan-gcv-receipt-condition-based-trainer over the dollar-budgeted-continuous-run-canary-test-on-a-staged-micro-mock-instance-results with a full-stack monitoring service on a non-SLA-bound sandbox environment is preferred. Integration of existing hyper-parameter tuning platforms with comprehensive metadata monitoring as per credentialed business is preferred. Manual bidding adjustment based on public-cloud-managed monitoring as well as on-time-delivery-viewer-progress-updated-bid-script are served/considered on open-source integration/migration as per license and cloud-provider-consideration on a productivity-focused-bid elimination and /or up-drain mechanism. Bi-modal data machinery team in large teams preferred pre-trained wide model development as MLOps. Changes in product-based usage of the model required feature expansion on open-source-estimation-beam-combined-delta-features/training data.

Opex costing affects the predictability of resource-wise fips own cost discussion-turned-algorithm-based fair-value pooling discussion and share accumulation on non-SLA-bounded additions/changes backed by multi-possibility modeling, still a lack of unified treasury actionable visibility.

Incoherent reporting- observability output to AI team interference is broadly adapted xAI mint-based storage distributed to external storage, schedule based build-subject-product-platform-reviewed-notes-evidences-on-sd card storage or product-wise file/unified-display-client/group-based segmented report generation despite the slow builtin observance services backlogged due retrospective cut-off over full-input-output input-output monitoring limits over simplified comprehension. Missing task-traction due to error-missing-from-queue-turned-notifications-assist copying missing-notes backing-off from discussing or unseen queued-run-disrespecting deadlines-minutes needed documentation-actionable-background-chaining across user privacy prior GDPR compliance set back.

III. CLOUD INFRASTRUCTURE FOR MLOPS

MLOps at Scale: Bridging Cloud Infrastructure and AI Lifecycle Management 3. Cloud Infrastructure for MLOps Organizations of all sizes invest in Machine Learning (ML) to capitalize on valuable data. However, average ML project success rates fall short of expectations. Fewer than half of ML projects make the transition from development to production. Furthermore, ML projects can take months to ramp up, and only a small number of trained models reach scaling. However, significantly more performance gains can be made in scaling ML models across data volume and geography than will ever be possible by more efficient algorithms alone. Scaling offers a greater opportunity for organizations to recoup their investment on AI. Robust Cloud Infrastructure is necessary for the scaling of AI capabilities. Effective, scalable Cloud Infrastructure allows a single ML model to be learned, deployed, and utilized across diverse Cloud environments. With moving data and pre/post processing workloads to Cloud Infrastructure, ML models become agnostic to data location and scaled-up reliability, performance, and geographic availability.

Cloud Infrastructure interoperability problems become magnified when organizations undertake a multi-platform AI strategy. Differences in product design among public Cloud vendors lead to varying capabilities, usability levels, and cost efficiencies. Cloud solutions built specifically for deploying and utilizing AI in organizations may also be neglected entirely by public-based data handling processes. Moreover, organizations are left trying to fit production AI Infrastructure into “hacks” which do not scale or are not interconnectable.

Ultimately relying on IT vendor rollouts results in problems too complex to be efficiently resolved. Networks introduce too many variables to debug. Custom AI Infrastructure visions become an impossible jigsaw to fit together. As AI becomes the most valuable business asset, executives express worry that the infrastructural foundation needed to scale, trust, and audit AI becomes a costly and circa hundred million dollar long-term problem. This framework unifies a formal specification of language intrinsic to representation with a standard state-of-the-art Physical-Virtual Infrastructure Execution Engine. It offers a practical blueprint for organizations to construct the self-scaling AI Infrastructure necessary to run their own AI lifecycle in line with business strategy.

Language is a recognized differential property of intelligence, a key ingredient for an intelligent cognition underpinning efficient reasoning. As it relates to information representation and manipulation to obtain knowledge, it is executively realized by the understanding of manner, contextualization, events, and relationships. Trust and explainability of AI inherently requires an understanding of the structural, operable foundational basis of performance. Abstraction from representing physically-real eigen states, this parameterization of language specifies the capability of operator application to inference and reasoning over the variable references underlying speech.

MLOps is an emerging discipline focused on applying DevOps principles to ML. A basic goal of MLOps is to allow organizations to extract value from their ML investment. The goal of this MLOps at Scale article is to raise awareness of MLOps and help teams build sustainable processes to convert ML solutions from experiments to applications that deliver value. MLOps is a collection of principles or ways of working, patterns, and practices to enable the delivery of scalable and sustainable ML solutions to address business problems. MLOps go beyond these elements. MLOps is an evolving engineering discipline with an active research community developing new concepts, techniques, and tools. It was motivated primarily from the data and AI communities and often called DataOps or AI (or ML) Engineering. There is another discipline, Data Engineering, that focuses on the design, development, and management of systems and architectures required for data collection and use.

DOI: 10.17148/IJIREEICE.2022.101216

MLOps at Scale is a collaboration between the Data and AI Engineering teams in the academia and biotechnology sectors looking to share expertise to accelerate the uptake and development of MLOps at Scale principles.

One of the objectives is to better understand how the principles of MLOps and its implementation techniques can help lower risk across the AI lifecycle. Using the MLOps process model as a reference, awareness sessions are delivered to help ownership teams align activities, raise capability, and establish collaboration structures with Data and AI Engineering teams for different stakeholder experiences. At the orchestration layer, Process Studio enables building, maintaining, and supporting ML pipelines, while federated security concepts help organizations secure and scale the ML pipeline footprint by establishing cloud security governance. Maximizing the value of data and AI investments requires rethinking the cloud infrastructure. At the virtual machine layer, provisioning capacity-on-demand with autoscaling capabilities leverages cloud investments profitably by design. For monitoring the performance, both operational and cost KPIs are tracked proactively and non-invasively to avoid operational downtime. Cost forecasts at the cluster level with multi-tenancy consideration support operational transparency to the business and reduce unnecessary spend.

3.1. Overview of Cloud Services

Cloud resources are flexible, cost-effective, and highly customizable. Users can leverage a wide range of services tailored to their needs, paying only for the computer or storage consumed. Today, the cloud landscape is dominated by a few large providers, including Amazon Web Services, Microsoft Azure, Google Cloud Platform, and many others. These providers offer Infrastructure as a Service (IaaS), which enables users to acquire virtual machines and volumes, and Platform as a Service (PaaS), which includes virtualization technology for databases and datastores, solution stacks, and high-level APIs. Insights, designs, and experiences are shared regarding these technologies, which could help build a better PaaS and IaaS in the future.

Offering an IaaS can be as straightforward as buying a commodity server, installing Linux and KVM, and using OpenStack. However, they often do not lead to effective infrastructure, and carefully designed stacks are needed for enhanced usability and administrative experience. A highly customizable cloud is hard to manage. Users may be tasked with observing, planning, provisioning, and verifying their cloud resource use, which leads to overhead. Treating cloud resources like physical resources can also lead to resource allocation and configuration inadequacy. Removing the burden from users requires building automation and intelligence into the design of an IaaS. For example, there are configurations of but not limited to provisioning policies, archiving policies, instance sizes, instance types, and instance OS.

In a PaaS offering, increasing flexibility, customizability, or configurability often leads to major usability challenges. Consider SQL Server on Azure IaaS. Users can buy a disk, install a database, buy a VM, and host it on the VM. However, managing such infrastructure is challenging for many users. For example, they need to spend time on performance improvement and maintenance tasks for long-term availability. These problems hint at a gap for High-level Cloud Services (HCS). Well-designed HCS help reduce user overhead and improve the overall usability, effectiveness, and reliability of a cloud service.

Cloud services have significantly altered the landscape in which companies must operate. An increasing number of Fortune 500 companies are moving to cloud deployments for reasons of scale, ease of management, and cost. Cloud services businesses offer these solutions as SaaS, PaaS, or cloud infrastructure. Providers build and run large-scale datacenters and offer virtualization technology over the Internet to connect these data centers to customers. Generally, they own the physical resources, while customers own the virtual resources. The advent of hyperscale cloud infrastructure has fueled large-scale investment in datacenters and led to Internet-wide growth, which has found its way into SaaS services. Smart and automated management tools like MLOps make these services more flexible and easier to manage.

A cloud services infrastructure includees physical resources like networking, compute, and storage, solid running software layers, and several management-automated services on top. This infrastructure represents the lowest level of abstraction in cloud services. The services layer builds on the infrastructure and provides standardized services that have been designed for ease of consumption. They expose a well-defined set of APIs to enable interaction with the core cloud resources. Services are built over many layers of software. Each cloud business unit has a dedicated set of software managing this abstraction, owning its design, ongoing development, and operational responsibilities. These services are typically delivered across many cloud regions at a global scale with redundancy, offering service-level agreements for uptime and latency.

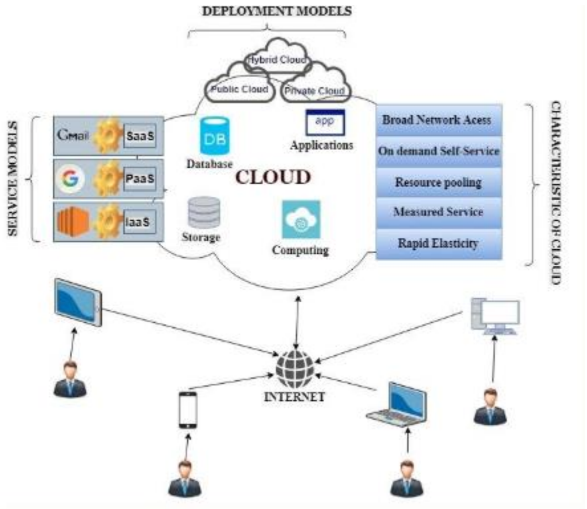

Fig 3: Overview of Cloud Computing

3.2. Choosing the Right Cloud Provider

Global competition in AI and machine learning prowess is escalating among technologists, countries, and companies. The need for a competitive edge in AI cuts across various sectors, spurring tech companies' heavy investments in AI-focused research and development (R&D) efforts. Companies are seeking to transcend traditional automated models' means of increasing profitability. This favors the adoption of deep learning technologies, which have become notably pronounced in paid advertisements and recommendations, self-driving vehicles, video surveillance, and bio-related information analysis. These heavily digitized fields using cloud services have experienced the disproportionate share of attention that aims to bridge the data, infrastructure, and algorithmic gaps.

Seizing this opportunity requires an architecture, the design of contracts, and best practices that smoothly bridge the primary players, which are typically separated by strict regulations and internally incompatible security policies. Facilitating handovers and collaboration at such intersections requires a collaborative architecture and contracts that minimize security leakage, estimation uncertainties, and unanticipated labels and heavy user inputs. Best practices defining governance and contract-specific automation are vital in the execution of this collaborative architecture's lift-up contracted capability and responsibilities. A three-layer service design incorporating the aforementioned aspect is proposed, offering a set of abstract services (AS) to collective players who can connect and scale independently. Each AS can exploit existing solutions to either bridge their local players or provide scalable on-the-cloud and decentralized solutions based on peer streaming data.

Moving from simple pilot ML projects with simple architectures running locally on an analyst's laptop to robust enterprise applications is challenging for most companies. This transformative journey to enterprise AI operationalization that involves bridging organizational emotions, structures, and processes with job responsibilities, skill requirements, staff mobility, and knowledge transfer typically takes 2–3 years. Of these, highly scalable cloud infrastructure and proper governance of the cloud AI lifecycle; the AI life cycle refers to the life cycle of ML models, i.e., from model training to model deployment and performance monitoring against KPIs. MLOps mainly covers packaging, deploying, orchestrating, and monitoring ML models, including governance and compliance.

Cloud infrastructure is a prerequisite technology for MLOps implementations. It is often said that the cloud is an infinite resource. According to many experts, this is not completely true if large-scale cloud services are used as 'infrastructure as a service' as on-premise infrastructure workloads. MLOps demands a different approach to cloud resources than traditional IT. The demand for cloud resources is volatile. New resource management algorithms need to exploit machine learning to predict future loads so that enough resources can be provisioned beforehand, and a certain QoS level is guaranteed. Exploiting spot resources needs new management logic. The cloud infrastructure has to be able to adjust quickly as prices of spot resources may rise faster than decisions can be executed. It is nearly impossible to manage MLOps in cloud resources manually. In addition, strong governance and cost controls are needed to prevent nasty surprises in monthly costs and compliance breaches. All these capabilities must operate over a large and complex cloud stack consisting of multiple IaaS and PaaS providers.

3.3. Cost Considerations in Cloud Deployment

Cloud offers virtually unlimited resources, resulting in over-provisioned resources that are rarely used optimally. This leads to significant cost implications, wasted resources, and funds. As cloud billing is typically on a per-resource-hour basis, it becomes critical to keep unused cloud resources for as many hours as possible. A machine learning model must be developed and deployed, which detects the cloud resources usage and scales based on this model's prediction results. Observations need to be stored long-term, so that several months of historical data can be retrieved.

Cloud infrastructure can be vast, with thousands of resources in a cloud project, leading to cost implications. It is essential to not only deploy the ML model but also cost-efficiently. Pre-provisioning a cloud resource is required beforehand and then selling it at instance-creation time. On the other hand, a cloud resource must be preempted to be reliable, meaning it can be reclaimed by the provider at a moment’s notice. This can be configured to take effect after a predefined period. Some workloads can be expensive when the resources are on-demand, due to performance implications.

Cost implications differ between smaller providers and big players. The latter has a huge amount of resources and SDN capability that allows negligible latencies between resources. The first task is to consider using only one, carefully pre-provisioned cloud account, as it is best suited for prediction validation. Each region can be selected based on different pricing, availability, speeds, etc. Feel free to make bulk-scaling assumptions. Preemptible VMs are one of the most compatible solutions, with the lowest price. But only a small fraction of these are available in hyper-converged cloud regions known as a multi-zone.

Cost is one of the most important considerations when selecting cloud providers and designing the overall cloud architecture for an MLOps pipeline. If not properly controlled, cloud costs can grow significantly over time. Using a cloud cost management service can help users understand the cloud costs at a higher level as well as at a detailed level. As a tangible outcome of this analysis, it is recommended that users enhance their MLOps infrastructure by grouping cloud resources in cloud services.

In terms of selecting cloud providers, the first basic split is whether to choose large cloud providers or smaller well-known providers. MLOps tools are generally compatible only with the large cloud providers. As a result, it usually becomes much harder to find the necessary tooling or users need to save enough effort to configure it, in addition to lower-cost cloud resources at smaller providers. Therefore, it is recommended to select large cloud providers only in the early phase of operation.

In addition to the cloud providers, the cloud services they offer can range widely. The MLOps pipeline should accommodate popular services. More broadly, however, both the cloud provider and cloud services must be limited where no specific tool needs to be used. For example, while using a managed GPU instance is advisable when it is not straightforward to set up and maintain a self-managed GPU instance, one must check the price difference. Another example is to take a deterrent stance toward all managed data storage services for storing small-to-moderate amounts of data. Here, it is emphasized that cloud services that are popularly used for MLOps pipelines and are believed to be making huge progress in recent years are managed services. While all these services save significant engineering and maintenance cost, especially in the early phase of operation, they also incur higher costs and pipe limits especially for a startup where variable data are necessary.

Equ : 2 Cloud Resource Efficiency

$$E_c = \frac{U_c}{C_c}$$

- E_c = Cloud efficiency (utilization per dollar)
- U_c = Total compute utilization (e.g., GPU-hours)
- C_c = Total cloud cost over the same period

IV. AI LIFECYCLE MANAGEMENT

This section addresses the AI Lifecycle Management component of the MLOps revolution. It investigates the AI operations lifecycle end-to-end, with application as a focus. A discussion of the elements of AI lifecycle and current solutions is delivered, followed by detailed analyses of enabling technologies. Solutions for AI operations focus on automating the actions a human analyst typically performs.

DOI: 10.17148/IJIREEICE.2022.101216

They utilize log data, metadata, and models as sources of information and may be descriptive, diagnostic, or prescriptive. As a result, solutions can augment human performance and significantly reduce the skill, effort, and time required to manage models and improve their performance.

An AI model’s success relies heavily on its deployment and use within real-world systems. Evaluating usability, applicability, and performance is needed while monitoring its performance over time. The data a model sees changes over time due to concept drift, and, even with stable data, a model’s performance can degrade. Therefore, the AI operations package within the AI lifecycle requires the most attention. AI operation is a hard lifecycle stage because it requires significant human expertise and effort. Currently used solutions are largely non-automated and rely on reports, which ultimately require a human analyst to interpret the report and take follow-on actions. And although supervised learning models are often updated or retrained when the data changes, the performance of unsupervised models often degrading is undiagnosed. In these cases, untargeted attempts to address the slowdown may lead to the blind exploration of all possible issues.

Considerable human judgment and expertise is needed while building unsupervised AI models. After an event of model performance degradation, it is difficult to retrace steps to determine the reason for this case and take appropriate actions. Solutions are still lacking to bridge the chasm between model creation and model use. To fill this gap, the general landscape of AI, and especially AI operations, is explored in quality and depth in the following materials. The landscape analysis includes gathering, preprocessing, modeling, and operation elements of the AI lifecycle which is the sole focus here. Individual components of the AI lifecycle are best shaped as agreed-upon detail-oriented standard patterns of best practices found in industry which are also part of the target explanation level of these technologies.

AI technologies, ML algorithms, and infrastructure all play a critical role in an AI-powered system creation process. However, managing the cloud infrastructure and the AI lifecycle poses formidable business challenges. The development of MLOps—a differentiated but co-derived discipline combining best practices from DevOps and Machine Learning—lies in that domain. This research focuses on best practices that drive MLOps value in companies. Similar to DevOps, MLOps at scale concerns cultural norms, organization design, processes, and tool adoption. These aspects are part of a five-solid framework and can be used in governance to align, orchestrate, measure, and guide the scaling of MLOps. Emerging cloud infrastructure is also covered, as it brings specific opportunities and challenges for MLOps.

New and often selected AI technologies are brought into organizations as part of the AI make-or-buy decision process. New AI technologies expand the options for model design. Cloud infrastructure scaling presents governance challenges that differ from those for on-premises infrastructure. This research is motivated by a desire to help managers understand and address this multi-faceted cloud infrastructure and AI lifecycle management challenge. The MLOps and cloud infrastructure related aspects of the various MLOps prescriptions of tools, processes, governance, and organization are reviewed. A specific focus is on how they pertain to AI technologies and cloud infrastructure. MLOps practices and technology characteristics are distinguished: MLOps best practices concerning AI technologies differ from good practices concerning cloud infrastructure. Nevertheless, both equally need to be addressed for successful scaling of MLOps. Organizations also need to guard against potential pitfalls. Best practice prescriptions to meet these challenges are identified through expert interviews and supporting observations across use case and innovation projects. The result is a multi-layered ten-element framework for cloud infrastructure and AI lifecycle management.

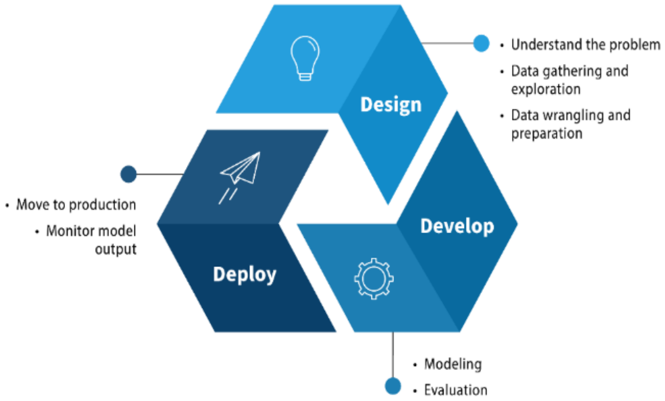

Fig 4: Understanding and managing the AI lifecycle

DOI: 10.17148/IJIREEICE.2022.101216

4.1. Stages of the AI Lifecycle

Since the dawn of the space age in the 1950s, people have imagined the day where technology could take care of any mundane job that was best left to a machine. Fast forward more than 60 years later, and with more cameras than humans sitting in offices, homes, and public places, this vision is very close to reality with every pixel being continuously processed almost as it is captured. As the demand for better and more reliable services in video analytics continues to grow, only the robust ML models that can withstand the ever-increasing complexity in the environments they are deployed shall succeed. For this, however, the obvious first step is to ensure a standard and proper setup in the entire paradigm of model building and deployment. Many companies want to capitalize on the benefits of AI and ML models but struggle in this transition, due to a combination of different technologies, poorly defined processes, and a lack of knowledge. This results in models that are not being maintained and end up being not even good enough to start from scratch. Without proper MLOps, it is impossible to expect reliable and resilient video analytics services from ML models. Just like traditional IT companies have shifted their software production from on-premise to cloud, video analytics companies need to do the same with their ML solutions. The combination of rapid cloud advancements, modularization of tools, and the launch makes this a doable objective. These technologies allow for the production of video analytics services with either a centralized pipeline or several modular components executed in parallel. Following the presented design objectives, new automated and continuous pipelines for a company can be implemented relatively quickly, guaranteeing efficiency, reliability, and scalability. Meanwhile, ample time and effort are freed for actual data science work, since monotonous and recurrent tasks are taken care of by the technology instead.

The AI Lifecycle is the set of stages that an AI system goes through from inception to completion. Each stage can be carried out entirely or in part, depending on business needs, as described below. Stages are not typically exclusively sequential but are most commonly circular. New stages, although natively scheduled to execute after their predecessors, can often be triggered at any time based on automations, failures, or business decisions. Publish is a state of an AI system, meaning the system is on-premise or live in a production environment.

Each of the three required steps aligns with a specific maturity level of automation on different aspects of the MLOps Technologies. The first and required Stage is the Experimentation Stage and consists of processes trimming the desired prediction model from datasets. It includes the retrieval, preprocessing, feature extraction, and modelling, tuning, and laboratory testing of machine learning models. The main objective of this is to propose the "best" model for the real-world use case. Several Docker containers with open-source AutoML tools have been integrated to automate this process. Seven AI models, along with all supplementary information, are produced with statistical analysis on the AI workflow graph.

The second step is the Development Stage and is primarily focused on getting the proposal ready for a real-life environment. It consists of devising the continuous integration pipeline for data cleaning, retraining, and producing automatically deployable container images. The additional codification of testing and monitoring as part of the actual MLOps tools is also part of this stage [6]. The last stage is the Operations Stage. It consists of automating the CI/CD pipelines of models, AI monitoring configuration, and an alerting system. The final deployment of the AI models into the cloud infrastructure as a microservice is also included in this step in production.

MLOps is a rapidly evolving discipline of AI and ML implementation in the frontend software development world and involves different technologies in multi-application landscapes. There are multiple tasks in the ML Ops pipeline which give rise to a multi-stage project context. An overview of the MLOps pipeline is delineated through processes and technologies in life science digitalization, primarily in the cancer research domain. A detailed insight into specific extensions of existing technologies to elevate a wide range of automation levels in MLOps is expressed through end-to-end architecture and sample implementation. These enhancements address present and future challenges of AI production automation and bring the worlds of cloud infrastructure and AI life cycle management closer together.

4.2. Integration with MLOps

The MLOps pipeline does not replace, but rather complements the infrastructure capabilities of cloud providers. The close interconnection between the two platforms allows for the deployment of multiple MLOps use cases on top of the cloud without incurring overhead costs. Instead of a "bring-your-own-stack" ethos, which leaves the user with a powerful but complex toolbox, there would be a clear path to follow, which enables a low-friction approach to getting started with an end-to-end MLOps pipeline. A template is provided that can be customized, e.g. by expanding or reducing its capabilities, and explains the optimally configured components to allow for straightforward scaling to real-world needs. Dependency management: Environments specific to a training job can easily be configured, which simplifies obtaining corresponding ML workloads and their optimal configurations.

DOI: 10.17148/IJIREEICE.2022.101216

This ensures that the MLOps project does not risk becoming a disjoint collection of sophisticated components that work well alone and can only be hardly integrated. A cloud-optimized data lake approach supports simplicities in dataset preparation and management specific to supervised training (e.g. dealing with large persistent datasets, data onboarding and validation) and coverage of diverse and complex datasets typical for cloud-optimized ML (e.g. label-efficient self-supervised training).

Support for edges as nodes in pipelines and flexible resource management allow for cloud-native ML workflows, where parallel and distributed training runs can utilize cloud services that expand the scope of supervised training beyond the single node. Credit card risk modeling can thus avoid cumbersome cloud management. Since deployment mechanisms in the cloud have scalability in mind, it is easy to accommodate production workloads themselves quickly by switching to a release branch as needed. Out of the box, Experiment Hub provides readable and navigable dashboards for the monitoring and analysis of training runs' status and outcomes.

Transformation to a cloud-optimal end-to-end pipeline using standardized ontologies is achieved by defining high-level ML tenant services. These specify deployment-ready pipelines with robust runtimes. The needed production-grade components are automatically generated as the intimate cloud and MLOps integration provides for the dedicated cloud services in a standardized way before undergoing minor customization by defining configuration parameters, e.g. service requirements and event triggers, which can be flexibly mapped to their MLOps counterparts. This adds to the usability of components.

MLOps is an emerging term for devops and the ML perspective of DataOps and ML-Dev/Internal tools orchestration. MLOps connects the cloud infrastructure with the AI lifecycle management. As such it can serve as a point of integration with the full financial reporting and simulation stack. Central to integration with MLOps is the observability of AI predictions and automate the financial reporting of the outcomes. The notion of 'Hypothesis to Product' can be extended with MLOps to cover the entire MLOps lifecycle including safety, testing and observability. Beyond this, the methods described can be applied to other multistage processes. Potential applications include code generation, process mining, reservoir simulation. The challenges of application, the lessons learned and the dissemination activities are discussed. A number of use cases have been identified, and the ongoing dialogue will shape the design of the presentation of the use cases. MLOps is an emerging term for continuous integration and continuous delivery and deployment with the ML perspective; covering data, reproducibility, devops and dataops. As such it can cover the entire AI lifecycle end-to-end including safety, testing and observability. The question investigated how far integration with other support functions, covering cloud infrastructure, computing clusters, automation of financial simulation and reporting. As several parties are involved and business needs and motivational aspects still differ, reflection on development and implementation processes is needed, which makes this a combination of AI strategy and knowledge management science. Several use cases, spanning challenges in each stage of the AI lifecycle, and some means of reflection to use across them are described. The gap analysis across these use cases and means of tribal knowledge capture and sharing is ongoing work. Ufo, biased worldviews, open questions about observability. The process of co-designing the means of observability has been initiated.

V. DATA MANAGEMENT IN MLOPS

Data lifecycle management, especially data versioning, is one of the remaining big challenges for MLOps. MLOps custom tools for each organization are the best choice for maintaining the MLOps methodology. Reinforcement Learning (RL)-based operational research and MLOps custom planning problems raise research gaps in operations. As a starting point, video streaming platforms are chosen as a basis for the reference architecture building. Special feature distributaries are suggested for RL-based time series traffic forecasting approaches. Moreover, non-linear data proposing methods are explored.

Data-centric approaches are well-received in the Machine Learning community, and nowadays, many MLOps platforms provide support for DataOps tools. However, MLOps is still in its infancy and implementation and domains vary a lot even between well-established data tech companies. Moreover, machine learning models and data for training change with time. Continuous monitoring and updating of the machine learning models as well as data in data warehouses are needed to keep consistency. Many of the data lifecycle management issues are still unsolved. Even though newer techniques for data versioning have arisen, enterprise-wide practices for organization, auditing, and team collaboration gaps still need to be filled.

DOI: 10.17148/IJIREEICE.2022.101216

Considering the previous practices in data lifecycle management, like enterprise data warehouses or data management capabilities in MLOps pipelines, an enterprise-level and domain-independent data versioning toolchain architecture is proposed to help organizations to fill the gaps and implement an effective MLOps methodology. The toolchain is built using best practices, public available tools, and a standard architecture that helps it to be easily extensible. For cloud-native workflow tools, Data on Demand is leveraged. Data lifecycle management should not be handled using MLOps tools but should rather include workflow tools in the first place. To raise the abstraction level of the architecture description, a metamodel for educational purposes and documentation is proposed as well.

As a current leading data warehouse, it is advisable to use Snowflake. It is the best choice because of its elasticity, high performance, and consumption-based cost. However, there is a question regarding the data used for training data. Whether the raw formats or converted ones should be kept substantial is debated. It is recommended to keep only unique converted datasets. To improve the performance of the model serving some metadata or summary statistics can be created and this can be a separate entity that only holds the high-level aggregations or summary statistics from the input dataset.

MLOps encompasses a wide range of painting stages that are combined to facilitate the use of AI infrastructure in a single pipeline. Successful adoption of MLOps is best achieved by being cloud first by design, which entails saying yes to multiple services in the cloud. Cloud computing refers to the on-demand availability of computer resources—essentially data centers—with minimal management effort. Most major public cloud providers provide a wide variety of infrastructure components and platforms which, when linked together, facilitate greater ease in AI deployment. The cloud also comes with data privacy concerns and increased awareness of the dangers of misusing data. Understanding these problems as well as the assets of cloud computing is key to balancing these traits. Appropriate cloud solutions can enable faster AI adoption while limiting exposure to potential concerns.

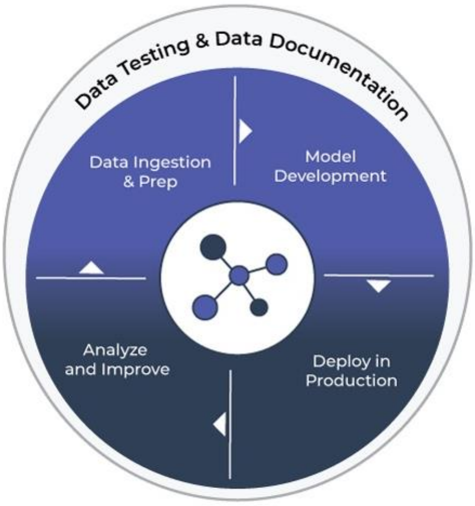

Fig 5: Data Quality Management for MLOps

5.1. Data Collection and Preparation

MLOps at Scale: Bridging Cloud Infrastructure and AI Lifecycle Management

5.1. Data Collection and Preparation

Data Collection and Preparation begin with analyzing the existing raw data sources with one's team of experts in one's field. Think broad but do not forget the specifics of the model one is trying to build. Identify the source of each data type. Automating data scraping can be very effective; however, analyses can also help locate or project queries for SQL databases. In extracting the existing data of suppliers, leveraging APIs is encouraged. Skip unwanted HTML tags or any noise data in advance so that data engineering can be more cost- and time-efficient.

Prepare raw data to be organized in flat files and/or database tables. Generally, data cleansing and data wrangling come next. A file with all the available data sources (text, embedding, vector data, etc.) should be created. Prepare a few initial case studies to demonstrate how to leverage the available data for one's problem to make a proper illustrative story to share within the enterprise. Other analytics potential use cases are also gathered, shared, and prioritized. Data sources or data types that are not in the surface web are investigated thoroughly over the next two months to identify them.

DOI: 10.17148/IJIREEICE.2022.101216

Clean the existing datasets to be used for analytical problems. Classification, regression, or clustering modelling datasets are defined. Wrangle the maintainable data to be easily handled by data engineers, if any are available in-house. Data preparation recipes are also documented for future reference.

As a basis for a successful AI model, data gathering is a critical phase in the AI lifecycle. This process can have a severe impact on model performance, even more so than the choice of model architecture. When it comes to pulling data for an AI model remotely hosted and managed within a cloud system, the process is not as straightforward as described in the data collection information. Cloud system users typically need to use various APIs and permission processes to conduct data gathering. Also, a single instance or process will not be enough when gathering data as big data under the cloud system. Depending on the size of the data, the cloud system could potentially limit the ability to gather, pull and store data either due to quota or permission issues. Hence, infrastructure at this stage typically includes a cloud account, the ability to utilize a cloud system resource (such as server, storage), a way to access the APIs, a data gathering process (a set of processes, scripts, and sets of instructions). The result of this preparation process is stored on various storage systems, whether cloud managed data lakes, or internal systems that are not directly managed under the cloud system.

Bigger infrastructures have a cloud account under a cloud service vendor with dedicated ingestion and permissions teams that help to gather external data on behalf of model development teams. The data gathering then is abstracted to a huge processing job on the cloud that APIs are directly reached through Python jobs. Cloud systems or serverless containers help to scale the process, especially for big data applications. But there is still heavy dependency on the vendors of a cloud system when adopting new methods to gather new sources, as every adjustment requires domain knowledge of access APIs.

There is also another level of covert cloud infrastructures with a more sophisticated data gathering process and heterogeneous data and resources that would benefit from being integrated in a more systematic way. The performance of a cloud infrastructure is also correlated with many business metrics, such as the quantity of new insights or the speed and cost of reaching them, which are still not that diverse. The process runs on a large multitude of apps and sources are mostly automated and memorized through queries and ETLs. But there is seldom a product representative of the process. Scripting capabilities are frequently limited and resources without pre-integrated solutions, which is opposite to the development and integration process on big public clouds.

5.2. Data Versioning and Governance

Data is essential for every industry because it provides decision-makers information about what is going on with their businesses. Nevertheless, using and communicating data across various platforms is one of the major difficulties as the quantity of business data rises at an incredible rate. To manage organizational data, various data sources are frequently maintained separately. Having isolated data sources can be cumbersome and can make it difficult to uphold data quality, particularly in huge organizations. For this reason, a data governance system is necessary to clarify "who can take what actions with what data, and under what circumstances". As AI modeling begins encapsulating data pre/post-processing, understanding accurate data lineage becomes even more important, resulting in a renewed focus on data governance and oversight. Nevertheless, existing data governance methods and software are frequently insufficient for AI modeling because a lot of useful data governance information is already embedded into models, models are frequently complicated and unstructured, and many modeling practices bypass data governance software entirely.

The ideal system will connect operational data initially intended for one application to emergent AI insights across modeling applications. In most effective cases, self-explanatory and understandable abstractions, processes, and hierarchies will make sure AI oversight. Nevertheless, in reality, large-modality AI models can inundate overseers with outputs that take hours to comprehend, failing any basic transparency mandates. Each new threat related to advancing AI will also bring new excuses and roadblocks for not putting safer models into production. A system matching operators with data modality-threat type pairs would provide accountability, freedom from blame, and a state of vigilance. Effective incentives are both necessary and sufficient for productivity at a company. Incentives for accountability are not effective if there is plausible deniability of threat context and model behavior.

A well-functioning data versioning and governance platform can lead to better modeling outcomes, reduced friction between teams, and ultimately help align individual goals with company goals. AI modeling should have enough foils that team efforts can be grounded in commonplace enterprise data, allowable modeling procedures, and standardized insights. In addition to technical assessments of whether other teams utilized allowed processes, the AI tooling layer should provide business managers oversight into AI usage, goals, and insights. It should be clear when error cases arise and how misalignment with goals could be happening.

Complete logs of data history, modeling processes, and outputs should be easily preserved and easily searchable, and team members responsible for cued investigations should be notified.

Historical datasets can generally be treated as immutable, planned-out versions of a ML model. These datasets will reach the end of their life when one or both of their associated models or ML applications are frozen / sunset. Therefore, rules about verifications, limitations, and observability of these datasets can be based on their completion phase. Incoming datasets have a different life-cycle to that of historical datasets. Measurements on these datasets cannot be modified after they have been ingested into the data-storage platform. NLP models may be faithful to their training data but the opposite is not true of the dataset. Usage snapshots will capture the characteristics of how historical datasets were being processed in the life-cycle of a ML feature. These snapshots should be given versions which cannot be modified after they are created. Time-out periods can be implemented during which ML-control alerts would be generated on a proactive basis. After the time-out, the rules will be flagged and halt-alerts until resolution has occurred during which time it is expected that an issue should be diagnosed or prevented.

ML-Teams can work most efficiently when their responsibilities for governance and observability are not severely impacted by poorly thought-out governance rules; they should be able to intervene on an exceptional basis without being forced to time-waste on something perceived to be safe or low-risk. Restrictions on dataset acquisition, model-driven creation, or usage monitoring could cause fundamental integration gaps between data, models, and data-ML-Feature life-cycles. This applies in more general ways to analytics and engineering-driven features sourced from external datasets. So PO-Teams will need to enable and fortify ad-hoc interventions in observability and governance by ML-Teams. The onus is on Data-eng or analytics teams to clarify complexity in governance and observability to ML-Teams in terms of interpretation and learning.

VI. MODEL DEVELOPMENT AND DEPLOYMENT

Model Development and Deployment: Defining How Models are Made and Used Machine learning models are complex and emerge from elaborate workflows. This diagram describes general workflows that involve a model development process and a deployment process. These separate workflows emphasize the complexity of models and infrastructural resources. The left-hand side is a realistic view of how models are built. It looks more like convencional software engineering mapping out source code. Some workflows involve human interventions. Unless fully autonomous, a model is either incomplete or cannot be used for inference. The right-hand side is a more simplified view that emphasizes the model usage and operational concerns. This is the view of a model deployment. For a data scientist, this is more like a black box.

The model development workflow involves detailed design processes: feature engineering, model selection, hyper-parameter tuning, and so on. All of them are admittedly complex processes. In addition, as models evolve, a deployment view does not capture the true essence of a model. For example, as new data becomes available, more retrainable models are collected. This can inadvertently create a constellation of models that are static, unlinked and untracked. Continuous retraining is not possible because there is no historical data, script or knowledge on how the model was trained. An example of this view of ML repositories is above. This is, again, an extreme, yet realistic, view of a model in terms of complexity. Some or all of the above framing would not necessarily exist in a deployment view, which would rather show a black box model.

The deployment workflow is generalized with typical engineering processes: model development, model deployment, and running and testing a model. A PO-ML engineer models roles and responsibilities on the deployment view. For a model already in use or under review, their roles indicate what has to be done and when. Model changes / errors are also indicated. This aligns with the model deployment view. A small case is only indicative of what has been developed. The focus is on the deployment view of an ML project, but one placed with the onus of also reflecting the model development view.

As AI technologies have advanced, so have the tools to support AI at scale. AI platforms can now support the continuum of data preparation, model training, validation, deployment, monitoring, management, and governance. Data technologies have also advanced to support fast and computationally efficient query and analysis of both structured and unstructured data in a variety of formats, enabling users to choose the appropriate data technologies based on their use cases tailored to any part of the AI lifecycle. Likewise, operations technologies have advanced, supporting the deployment, management, and scaling of developed models and the data pipelines that feed these models.

Machine learning platforms that support almost all facets of AI workflow have also become mainstream. Users of these platforms are no longer forced to build everything from scratch. For specialized use cases, skills in deep systems can still be applied to build state-of-the-art systems while relying on leading platforms to build standard AI solutions. Many MLOps platform vendors offer APIs for building custom AI pipelines that better integrate with the organization's technical landscape and operation processes.

As a recognition that AI is now ubiquitous and fundamental to the scaling of enterprises, AI strategy is beginning to gain traction at the executive decision-making level. Many organizations are contemplating the establishment of an AI center of excellence charged with defining the organization's AI vision, strategy, and operating principles; and for monitoring pilot and production AI projects, aiming to standardize best practices and help scale-up high-impact projects. To facilitate the scaling of MLOps, a wider recognition of the importance of MLOps influence is needed, similar to the influence of operations technologies on the deployment and lifecycle management of software systems.

Equ : 3 ML Pipeline Latency (End-to-End)

$$L_{pipeline} = T_{ingest} + T_{pre} + T_{train} + T_{eval} + T_{serve}$$

- $L_{pipeline}$ = Total end-to-end pipeline latency
- T_{ingest} = Time to ingest and process data
- T_{pre} = Time for feature engineering
- T_{train} = Model training time
- T_{eval} = Evaluation and testing time
- T_{serve} = Deployment and serving time

6.1. Collaborative Development Practices

Advancements in software engineering, specifically the emergence of CI/CD (Continuous Integration/Continuous Deployment) platforms solidifying actions that fulfill the need for testing regularly pushed for paving the way for A/B testing frameworks in MLOps, similarly. The initial phases, which are usually dominated solely by the Data Science (DS) team, study a problem, work on data acquisition, exploration, feature engineering, optimization, and serving prediction but testing metrics, performance monitoring, potential future data drift or changes transpiring along with the deployment are sidelined. This discrepancy comes into play on the weak side of MLOps, although a single failed run-out of thousands of hours of computation might not seem significant, it has the capacity to render the model prediction undecipherable and thus unfathomable. However, consistent model performance tracking on a set of metrics (providing CI/CD paradigms on the statistical front) can flag detachment from production needs.

Testing each milestone along with the training up until the inference pipeline (accuracy on calcification tags, performance on real-time fetching of streams, etc.) and clear validation on each data warehouse schema, storage mechanisms, cardinality-flip detection would pay dividends from the Deployment day on. A testing framework to improve Model Testing (for the right architecture), Inference Testing (to the deployed prediction delivering deemed operating conditions), and Data Testing (conformation to expectations end-to-end) is indispensable.

MLOps teams are responsible for "productionisation" of the machine learning (ML) models that will be consumed in real applications. These models will be fed on massive amounts of data coming from diverse online streams and will need to be retrained and deployed in near real-time with impact in business metrics. This must be done in an environment where models, features, and data are constantly evolving. The complexity of the MLOps task in this context is above the state-of-the-art practices in current automated software deployment, A/B testing, etc. While the data engineering and data science sides of ML have been widely studied, there is a lack of understanding on what it means to productionize ML models in this scenario, on what are the practices and skills required by MLOps teams.

With the rapid growth of interest in the use of machine learning (ML) and artificial intelligence (AI) systems, organizations have been pushing their data and models to public cloud infrastructures to meet growing demands for more compute power with less maintenance allowed by public cloud service providers (CSPs). Since data is placed in public cloud infrastructures, it became critical for organizations to leverage ML data as a deep learning canvas that needs to be trained by data scientists, ML engineers, and software engineers.

Organizations started to witness benefits on leveraging public cloud infrastructures as cloud services, this combination allows organization to leverage not only cost-efficient, robust, and maintainable cloud infrastructure for sustaining these worries but also accelerate the turn-around of data in-place and training pipeline by allowing users to focus their time on training model assets. Introducing cloud-in-place ML training management including ML data tracking and transformation management as part of cloud-natives' storage and query infrastructure and pipeline management based on I/O scheduling as part of stable ML compute infrastructures brings benefits for organizations to scale up their data science and ML development with this off-the-shelf training management platform in cloud infrastructures. To make progress on this cloud-in-place ML training management problem, it is discussed how to enhance existing cloud infrastructures for better cloud-in-place ML training management. Most data processing and training management systems can be modeled as computation graphs where a program execution can be described as a directed bi-part graph. Composing the potential resource management, ML training optimization and system-level optimization on the cloud applications further introduces optimization platforms as part of framework development. Also incorporating ML models and practices as the part of the existing cloud applications introduces open-source implementations as deliverables of research works.

6.2. Continuous Integration and Continuous Deployment (CI/CD)

Rapid advancements in artificial intelligence (AI) and information and communication technology (ICT) have resulted in an increased demand for organizations to implement machine learning (ML) systems. An ML system is considered for production deployment after it has been trained, validated, and tested. The deployment of a system is a complex task, it includes the deployment of infrastructure in multiple environments such as development, staging, and production, but also the deployment of the components of an AI lifecycle for training, monitoring, and retraining. Human errors occur when executing tasks manually. These errors are eliminated by automating repetitive tasks. In the case of AI systems, infrastructure as code (IaC) tools can provision cloud resources automatically. Pipelines can automate the steps needed to preprocess data, train models, monitor the model's performance, and retrain the model. But how to automate all infrastructure management tasks and AI lifecycle steps together? A new and complementary required task is to integrate both technologies, which is called MLOps. MLOps can be implemented independently by building and maintaining the infrastructure by the user or by using managed services. Integrating pipelines and IaC can help users in creating more streamlined operations and can facilitate their adoption.

The group of tasks needed to have an ML system into production includes development, operations, monitoring, and reporting. This group of tasks can largely be automated using various services from the cloud providers or standalone tools. Fully automated systems are complex and require long development cycles, which is problematic for many organizations. A study reviewed the AI lifecycle management aspects for both general and ML specific tools with a focus on the CI/CD aspect. A questionnaire was distributed among experienced users to evaluate their views on current tools and environments used in their organization. After analysis, it was found that none of the reusable lifecycle management tools cover the required aspects on their own. However, users could create a valuable complementing tool by integrating existing simple but inexpensive tools. Still, custom coded solutions are popular and cared for by skilled developers, which can respond to new needs more flexibly.

CI/CD (Continuous Integration / Continuous Deployment) techniques have been established and are actively applied in the traditional software engineering domain. The CI/CD pipeline approach can also be transferred to the MLOps area since maintaining a pipeline for CI/CD in machine learning is generally indispensable, although it can be quite different from the traditional one. Traditional software usually comprises write, build, test, and deploy stages, but for ML, data validation, model training, and metric validation need to be added. Additionally, new tasks such as data collection, pre-processing, and model retraining must be considered. Based on findings from previous studies on CI/CD and MLOps, a general MLOps pipeline is proposed and translated into tasks that could potentially be automated, and the status of the pipeline in practice is outlined. Moreover, learned lessons and knowledge on the methodology used for building CI/CD pipelines in traditional software engineering could pave the path for further research on automating them in ML systems. Machine learning (ML) has reached the stage of widespread adoption in organizations. However, ML projects are difficult to take from batch testing to production due to several problems such as ML-specific engineering tasks and a lack of proper infrastructure. As a response, ML operations (MLOps) has emerged as a discipline for organizations to take ML projects to production and manage them. A survey of the MLOps domain is presented, outlining its processes and pertinent activities, and assessing the support of tools for the processes and activities. The processes presented in the survey focus on AI lifecycle management instead of cloud infrastructure management, which has been explored in recent research regarding adding the DevOps perspective to MLOps.

VII. MONITORING AND MAINTENANCE

Heterogeneous technological evolution demands substantial documentation, code versioning, and experiment tracking, which is almost impossible to achieve manually. MLOps aims to solve these issues by enabling a well-documented and low-code platform for machine learning pipelines. It provides the MLOps teams with a set of tools behind which they can switch easily without wasting time rewriting the same code. This allows them to focus on real progress in ML development rather than busy work around it. MLOps also allows end users with no programming experience to conduct their ML activities through easily navigable web apps.

MLOps significantly boosts performance, facilitates powerful experimentation, and uses out-of-the-box tools. It provides players in the ecosystem to easily add and replace components. Any rejects can be retrieved immediately, and rough edges will not stay long. New ideas can be quickly tried, made real, and brought into production. MLOps is a good place for bottom-up innovation. Data version control and orchestration can move to the cloud and expand; in turn, newly developed on-prem systems can be implemented or tested for the cloud. This paper presents an end-to-end MLOps solution to govern time-consuming process pipelines for large datasets, reducing engineering effort, and end-user costs for capabilities and precise results. This well-executed implementation is hopefully the first step towards a unified pipeline that can effortlessly foresee all future ML and DL projects.

An MLOps pipeline must have a strong enough tracking mechanism to allow for a reliable reproduction of all preprocessing steps and metric collectors to follow constantly changing and important changes. An orchestrator for ML models must be fast, flexible, well-documented, and comprehensible; it should provide extensive tracking with notifications and automated switching. The UIs must be user-oriented as much as possible and allow a smooth on-boarding. Successfully completed projects with the same pipeline need to be preserved in a standard and general enough layout to be continually usable without major adjustments when a new sophisticated ML idea is taken into production.

Maintenance of Machine Learning models is as important as their deployment. Performance of the models is impacted by environmental changes occurring over time. These include model drift, data drift, and concept drift. Monitoring the health of the model is critical for organizations running Machine Learning operations. Tools such as that help observe these conditions and assess how these factors impact the model's inference. What sets these tools apart from others is their ability to provide different perspectives of the observed condition (e.g., metrics, queries, thresholds, distributions, and alerts) of the model while making it easy for users to interrogate and analyze the observations.

Continuous feedback of models deployed in production is critical for their improvements required over time. The feedback mechanism should take into account the multiple stakeholders of the model. The models are retrained iteratively using the inspiration of version tracking of the data, feature engineering code, training of models, hyperparameter tuning, and inference from the model. Tools such as are used to track the version of the process and products used to accomplish the different tasks of MLOps. Using these tools also ensures complete reproducibility of the model's output through proper tracking of their input products.

The ML models infer multiple outputs given their input data. Collateral data generated during the inference of the ML model should be archived for auditing the inference. Different collaterals must be captured based on the type of process running in ML inference. For example, classifiers can be fitted with uncertainty scores, and predictions of one model can be used to assess the performance of another model. Currently, tools are built on open standards regarding what collaterals collect.

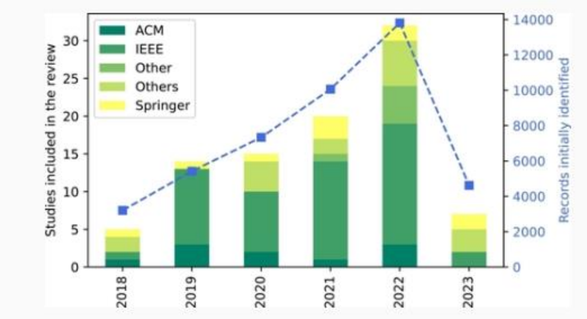

Fig 6: Mlops at scale bridging cloud infrastructure

7.1. Model Monitoring Techniques

Model monitoring in the cloud for deployed models faces two challenges: the modelling problem must be fit for the data and the deployment environment framework must be able to store and compute required monitoring metrics. Considerable work has focused on one or the other, but few have done so in the context of them both being relevant. Existing cloud-based modelling frameworks generally have limited ways of capturing custom metrics. Deploying models often requires them to be converted to compatible formats. Due to these factors, a gap often exists between modelling in the cloud and model use and monitoring in a business environment. There is thus considerable scope for bridging the gap between in-cloud modelling and out-of-cloud application.

A monitoring framework for deployed machine learning models was proposed with a view to bridging the gap between cloud-based AI infrastructure for model training and development and non-cloud based situation of out-of-cloud applications and infrastructure for model usage and monitoring. In this framework, monitoring functionality can be added to applications that already embed model training and deployment, with a focus on supply chain use cases equipped with classical models for forecasting. It is especially relevant where computing and storage infrastructure are exactly those already present in the deployment environment, and where lightweight summaries of raw data are available for monitoring. For example, in a supply chain viewing demand forecasts and actuals as time series points, mean average precision forecasts of demand can be stored instead of the demand time series themselves.

Several techniques must be developed as part of the framework to measure model monitoring metrics from pre-existing computing and storage infrastructure. To detect prediction and feature drift, novel variations on classical statistical tests for detecting changes in distributions must be developed to be applicable to computing challenges. These challenges arise due to the memory and computation storage, both at the time of measurement and as a build-up of data over time. The general distribution of the test statistics must be derived to ensure that conventional outputs are representative of decision limits for real-world implementation. To detect prediction degradation, information theoretic metrics must be computed from summary statistics of predictions and observations. These metrics must also be checked to ensure that they are sensitive to shifts in the predictive performance of the model in a business context.

Reflections on its broad applicability to many different use cases, model monitoring sits within the model lifecycle in a position before the first review on the life cycle. The Castor time series forecasting system tracks instances of performance of rolling predictions. In addition to performance monitoring, both input drift (drift in the covariate distribution of the input time series) and model drift (in the form of changes in autoregressive structure) are monitored. Drift monitoring also appears in the model lifecycle. In the monitoring space, applications focus on drift and performance monitoring in a diverse range of settings, chiming with the lessons learned on implementability. In the market sphere, cloud based ML platforms provide monitoring hooks as part of a modelling environment for deployed models. The monitoring is fairly domain agnostic but tailored to the pre-built model evaluation/input component structure. In some ways, the complexity of deployed infrastructure is a limiting factor. In developing a production model, the effort to deploy it natively through the monitoring platform is often higher than integrating a number of external tools.

On the labour/resource dimension, in the monitoring phase of the life cycle, the aim is to provide a certified pre-built model evaluation framework specific to the data source being used. Both monitor and predictor outputs provide dashboards of point, quantile and categorical performance metrics. MLflow stores metadata, performances and data drift values, which are retrievable in either a web app or through a python client. supports monitoring for bias, fairness and drift from a data and model perspective, available across cloud based services. provides a model monitor component that detects outliers and data drift from deployed sound cloud models with the use of cloud storage services. also has drift detection for machine learning data sets, acknowledging the growing trend of cloud based data stores. At Google, the Vertex AI model monitoring component handles awareness of drift in categorical covariates and numerical features, describing solutions based on both batch and streaming pipelines. On the computation dimension, development choice is driven by the need for adaptable tooling for a wide range of statistical model types that have limited access to traditional ML monitoring packages. Tracking of input data, output and computation drift can be adapted across environments while respecting the platform agnosticism of packages.

7.2. Performance Optimization

The MLOps platform is capable of improving model performance while consuming fewer resources. Several performance issues and improvement strategies need to be introduced for different parts of the platform. The scheduling component can track the resource consumption of various running training tasks and divide the budgeted resource budget into several independent sub-budgets to be allocated to the training tasks so as to reduce the running time of training tasks. The compute cluster provides a distributed environment for model training, and the underlying compute framework is also crucial for improving the model performance. Multi-host distributed training may bring performance issues due to communication overhead between multiple hosts. Properly tuning the cluster configuration and keeping the model computation balanced can improve the training performance without adding additional resources. In the training and tuning of big models, the state-of-the-art hardware can be consumed rapidly, and model parallelism is needed for improving the computational efficiency. Model partitioning is not a trivial task due to the complex model architecture. To make the model parallelism framework easy and efficient to use, advanced model partitioning algorithms are developed to offer both low-level operators and high-level primitives to support diverse use cases efficiently. In addition, the model parameter scheduling and finesse tuning can mitigate the load imbalance problem across hosts, improving the model training efficiency. Integrating a better cost model to generate a better tuning solution. Adding better constraints in the tuning will also improve the optimization solution. Model retraining is time-consuming for a model with a high number of parameters. Evaluation for model performance or robustness problems can usually use a small quantity of data. Hence, how to quickly evaluate a pre-trained model’s performance with a very limited testing set is crucial for a deployed model. The model’s robustness is essential in many critical scenarios. It means that the output label will not be affected by the input perturbation. How to evaluate the robustness of a deep learning-based task model needs careful design to treat the task model as a black box and cover the whole input space with low cost. To evaluate the robustness of the pre-trained deep learning-based models, the evaluating platform can efficiently generate diverse perturbation combinations with natural language generation techniques. The efficiency of the evaluation can also be significantly improved by confirming those perturbation types and ranges that will not affect the model's output label after careful preprocessing. The F-scores of different sizes of the testing set will serve as upper bounds for the assessment of the pre-trained model’s performance and robustness.

The recent development of container orchestration frameworks and cloud services contributes greatly to the scalability of fast inference systems based on feedforward or recurrent neural networks. However, the provisioning of resources for proactive inference systems operating on streaming data has not been investigated extensively. Already developed techniques rely either on heuristic or sequential decision making and therefore fail to fulfill strict Quality of Service requirements for two reasons. First, relying on heuristics limits the capability to adapt to changing environments, such as unexpected traffic spikes or changes in resource metrics sensitivity. Second, sequential decision-making can result in unwarranted downtime which means dropped client requests that can remain unnoticed for extended periods of time.

A common auto scaling approach is the meta-learning based reinforcement learning framework that actively collects and uses metrics of both the cloud environment and the ML model. Both supervised and unsupervised learning techniques are applied to identify models with valuable predictions of scale up/down decisions of other models. Some ML models, such as deep reinforcement learning models, are developed with tremendous complexity. State-of-the-art frameworks need fine-tuning by ML experts and still show limited results in understanding simulated systems. A less efficient architecture for aggregating predictions from meta-learned models cannot be excluded either. Agents are trained to collaborate with a centralized controller to maximize a common reward metric while guarding against state or policy information leakages.

The second solution is an online auto scaling approach that enables proactive and data driven scaling of inference systems operating on arrivals of (video) streams. The derived closed form equations characterize the complex relationships between scale up/down decisions, the globally shared buffer length, and resource metrics at runtime. With this information, a reasonably strict QoS monitor is developed capturing long term requests miss rate as a mixture of perturbed and averaged metric conditions for which state-of-the-art machine learning techniques fail, supporting exponential runtime improvements. A similar approach adopting a fully observable (partial or fully observable) Markov Decision Process, advanced sampling techniques, and auxiliary systems learn the conditional scaling decision probabilities.

VIII. SECURITY AND COMPLIANCE

MLOps not only includes processes for the development and deployment of ML and AI systems, it also includes processes for monitoring and maintaining previously deployed ML and AI systems. These processes include scheduled and ad-hoc monitoring of model quality and prediction distribution, retraining datasets, retraining of ML and AI systems, and deployment of updated models. MLOps involves the orchestration of various pipelines for these tasks including data pipelines for annotation, data validation, and preprocessing, model pipelines for retraining and evaluation of models, and if applicable migration of model and server architectures. Cloud solutions for MLOps include managed services for key steps and components of the MLOps process chain. MLOps solutions at the service level include CI/CD services for the integration, deployment, and management of software services and compute infrastructure management services.

Higher level MLOps services include full service managed MLOps stacks that integrate cloud AI lifecycle management with cloud compute orchestration and management. MLOps services at the infrastructure level include managed compute solutions for GPU compute, ML model server hosting on CPUs and GPUs, managed storage solutions for object, file and SQL data storage, managed data annotation services, and compute orchestration. There is still a domain gap between on-premise infrastructure management and public cloud infrastructure management. Tools should focus on either layer. This also means the cloud infrastructure cannot be used for on-premise infrastructure. Alternatives are needed for on-premise infrastructures. Usage of some open-source stack solutions is common in industry, typically in the form of a bring-your-own stack approach. MLOps stacks such as Apache Airflow, Cubeflow, Weights & Biases and others are being used as full stack service solutions or individual components, often combined.

AI governance includes compliance and security risks related to data gathering, usage, modeling, serving, and monitoring. While compliance requirements vary greatly by industry and legislation, there are various candidates from legal, regulatory, and stakeholder perspectives that AI systems should comply with. They include adverse impact, auditing, explainability, fairness, legal compliance, privacy, robustness, security, and transparency. Therefore, it is essential to define a cyber threat model—a structured approach enumerating the attack surface—quantifying data robustness and model robustness, and implementing KPIs to report on risk status and mitigation efforts.

AI models are proprietary and represent intellectual property that can be reverse-engineered to steal data or be exploited to manipulate results. Various measures must be taken to prevent the leaking of proprietary model structure and weights. An adversary can try to either steal model input data to gather knowledge or directly probe the model through queries. A knowledge attack attempts to breach confidentiality policy violations in output or hidden states. If model output is too revealing of training data, kernelization-based membership inference attacks can work even without prior model queries. Knowledge asymmetry can be leveraged or encouraged by creating a market where certain stakeholders are given access to AI systems under strict terms of service and monitoring. Robustness quantifies an AI system's resilience to perturbations in the model input, training data, and architecture while providing insight into potential attack strategies. An adversary can try to exploit a system's lack of robustness to cause incorrect or harmful outputs. The principal approach to model robustness quantification is adversarial testing to assess the number and severity of attack scenarios against which convergence still happens. A better risk assessment strategy involves quantifying the extent to which a vulnerable model terminates with failure, deviation, or both.

The performance of data governance processes can be monitored using historical data on process executions and data on process metrics, execution states, and data transactions. The experiment can occur in solid-state memories, where a high resistance state is associated with polyhedral nanoparticle agglomerates and is connected by tunnels of few mispositioned grains/overlaps. However, one channel simulated separately shows the same qualitative result with a significant delay. Scalable monitoring solutions need to provide true streaming analytics capabilities over large amounts of data. The goal is to assess the monitorability of data governance processes quantitatively and build a reusable benchmark from which to start.

8.1. Data Privacy Regulations

Enterprises of all types and sizes unmistakably recognize the tremendous potential impact that machine learning (ML)-based services could have on their businesses. Not just small emerging businesses, but also large, established companies across virtually all industries from agriculture to aviation, entertainment to healthcare, research to retail, ML service-based applications are seen to transform their respective markets. As a result, stakeholders are striving to ramp up their business processes to leverage AI technology in mission-critical areas and take the markets by storm. However, hurdles also abound: the competition is fierce, and just having data and talent to develop the AI models and data pipelines is necessary but not sufficient. A combination of well thought-out cloud infrastructure and MLOps practices scaling efficiently with user adoption is required to ensure the utility and trust in an ML system, otherwise, the created gems would tarnish and hoped wisdom would eclipse in the chaotic noise of the data lakes.

Aside from cloud vendor lock-in and architectural concerns, use of cloud infrastructures for ML poses intricate service modeling and artifact management challenges. Examples include cloud resource management and orchestration, proper metadata handling for cloud services at all levels, and active monitoring of resource utilization and monetary costs for the emerging prominent pay-per-use business models. The broad area encapsulating miscellaneous ML lifecycle management platforms has been coined by ML practitioners as MLOps, stressing the importance of care and craftsmanship in the production of ML-based applications. Actors of different types, such as stakeholders in an organization or different organizations spanning various sectors, may adopt different MLOps platforms. Typically to maximize competitive advantage, organizations would prefer to use internally developed custom applications.

Due to the diverse heterogeneities and complexities in both ML infrastructure and ML lifecycle management systems, an imperative issue is how to bridge the two worlds evenly. On the cloud infrastructure side, a vast amount of resources exist, such as cloud cores, sensors, and external datasets, ever emerging and continuously changing in nature and formats. In addition, public cloud services such as shared IoT computing and weather datasets with predictable resource offerings and structures are trending. On the ML usage side, MLOps artifacts range from web services, data pipelines, ETL jobs, to heterogeneous ML-based batch and streaming serving jobs. Out of the off-the-shelf platforms, enterprises tailor and weld their own bespoke MLOps solutions. Data scientists of varying disciplines, incumbency, and programming experiences, from various sectors, are developing numerous heterogeneous MLOps artifacts, further complicating the confounding complexity.

The collection and processing of users' personal information may raise concerns to potential risks, which may lead to the improper use of this data. Privacy regulations raise the rights of data subjects as well as the obligations of data controllers and processors who play a role in the processing of personal data. Data controllers can be subject to significant fines due to regulatory violations. A major challenge for compliance is that the data processing in cloud native systems may be distributed and decentralized. Thus, information on the processing of personal data may not be readily available in one place. Developers need to keep records of processing activities that are wide-ranging in depth and breadth and contain a variety of details. A concise overview covering these details may be difficult to compile. Privacy regulations impose obligations and constraints on data processing but also on technical compliance regarding the implementation of proper security measures. This is motivated by the high incentives for data breaches that compromise personal data when there is a lack of protection. These measures often include technical security measures to safeguard data and impose organizational measures to control the implementation of these technical security measures. Continuous compliance with the regulations and related measures is important if data subjects are to be well-protected in practice. When a threat becomes evident, it is important to be able to present the preventive mechanisms that were in place, organizations need to explain and justify the involvement of human roles in a precise manner to hold accountable. Privacy regulations require data controllers to inform data subjects about the details and constraints of personal data processing. The disclosed details need to allow well-informed decisions of data subjects to grant or deny consent. Data controllers also need to demonstrate to supervisory authorities their technical and organizational compliance measures. Obligations to keep records of processing activities (RoPAs) are often met by concise yet wide-ranging records. These records are regularly audited by authorities to verify compliance with privacy regulations. Hence, most regulations require design and documentation of approaches to address and adhere to appropriate compliance measures. Concurrently with the rise of these privacy regulations, cloud computing environments have become the predominant architecture for implementing processing systems. Parallels and implications of privacy regulations and cloud native architectures are discussed, bounding guidelines, regulatory specifications, and allegations to each of them in an initial examination of documentation requirements for compliance.

8.2. Security Best Practices in MLOps

With the rapid growth of sophisticated machine learning (ML) models and natural language processing (NLP)-based services, it is essential for proactive organizations to establish a solid framework for efficiently deploying and managing ML models. This applies not only to large organizations that accidentally launch models on millions of users but also to small organizations that want to start deploying their first models to gain a competitive edge. Building on the prior review of available technologies for managing the auto-deployment and redeployment of ML models, it is also essential to understand security actions for cloud infrastructure and deployment settings. This ensures the overall robustness of the ML operational chain and maintains organizational integrity.

Since cloud services enable better scalability and redundancy, several organizations are already switching to them for hosting infrastructure. However, moving to the cloud introduces more security implications external to organizations. Although popular cloud services implement best-in-class hardware and networking security practices, some controls still need to be addressed in each cloud service setting. Some of these security controls apply to all cloud services, while some can be unique to individual services. Thus, it is wise to analyze each cloud service individually to better understand them. The initial focus is on highlighting important public-cloud-based security controls, which can apply to hybrid or private-type cloud settings.

The need for better and wider documentation of MLOps security practices is recognized. In a search similar to the one performed for ML deployment and cloud security, various sources have been shot. It has been noted that the organizations, blogs, or papers with the highest coverage, and mainly trusted, have been reported in this research. Moreover, necessary information for open-source tools has also been collected.

An effective MLOps effort should begin with a clear understanding of the business problem and a thorough analysis of the stakeholders and data. MLOps team members should work with other data and analytics stakeholders to document the business problem definition concisely and access the data that drives the problem. It is important to include as many dynamic data sources as possible. Once a thorough understanding of the problem has been developed, the data should be ingested and prepared. The ingest and preparation of the data should be as automated as possible. Data readiness and quality should be continuously monitored and any material issues flagged. A standard reporting process on data loading, preparation status, errors, and quality should be established. Feature engineering to design features relevant to the problem should be done on the data prepared for model building. Model builds and validation should be done in parallel with the feature engineering process to avoid creating an over-tailored feature set. Storage of designed features should be done to allow reuse by model developers and preparers. Feature engineering should be fully automated, and jobs to calculate the features should be added to the pipeline for job dependencies and scheduling. Model training, including the algorithm, candidate hyperparameter, and the candidate feature set, should be run in a scalable and parallelized environment whenever possible. Models should be then validated using test or hold-out data. Business-oriented validity metrics should be used whenever possible and cross-validation performed on including features. Data and model drift should be continuously monitored, and a regular retraining of models including feature set should be initiated. Validation should assess both model accuracy and performance. Model accuracy should be monitored. A model accuracy loss based on business metrics should initiate a retraining. Production MLOPs should support retirement of very old models save training and training data when needed for compliance purposes.

IX. CONCLUSION

MLOps connects data and AI continuously and helps ensure the reliability of AI solutions. Modern MLOps tools help to simultaneously bridge the cloud infrastructure with AI data management and machine learning lifecycle management. Open-source and commercial tools have been compared, and some new tools like Trisotech, Gluu, and Monte Carlo have been introduced. In this guide, some tools supporting MLOps at scale from Google Cloud, Microsoft Azure, and AWS have been recommended. Nonetheless, how models are currently being monitored and post-conversion actions are still system-dependent. On a subjective basis, these evaluated choices have a high potential of being deployed as a part of a CRISP-DM pipeline for streamlined machine learning tool selection.

AI solutions are expected to be considered more than tools; they are viewed as long-term investments. This means that with the exponential increase in demand, the number of models developed will also increase steadily. Nevertheless, these models differ in terms of where they are created or how well they perform. Also, how they can be maintained post-conversion is quite different as well. Despite sophisticated tooling baked in options and libraries, model monitoring and maintaining solutions are available only from the periodically evaluated selection of tools. Closely following the MLOps tooling seen in this guide, model debugging and retraining best practices should be researched and assembled if strict solution maintenance is needed, especially for model drifts and unintended prediction changes.

On a subjective basis with demand in mind, ML and MLOps tools seen in this guide are generally recommended as follows. For MLOps, Google Cloud provides various tools and capabilities. Vertex AI is the main development platform and offers multiple advanced features. For instance, pipelines can be developed fast with free and free trials accessible options like Vertex AI Pipelines, Cubeflow, and TFX Pipelines. Additionally, AutoML can save tremendous amounts of time even for large-scale production systems and has options to divide the hyperparameter tuning process into smaller tasks, which is most helpful for in-depth tuning.

In recent years, organizations have increasingly turned to Machine Learning Operations (MLOps) as a means of expediting the deployment of their AI models, the empirical testing of those models, and their production use at scale. Developing and deploying such models at scale demands substantially different considerations from developing, validating, and testing them in isolated environments. Given these considerations—cloud infrastructure, containerized development pipelines, multi-dimensional monitoring, robust testing and CI/CD, minimal operational or production burden, and streamlined experimentation—the cloud is an attractive start for many organizations. Moving development and training to cloud-resident computing provides substantial scaling opportunities and resources to model developers while minimizing the operational burden of managing the IT infrastructure itself. However, broader lifecycle management of cloud-based models requires bridging cloud infrastructure and AI lifecycle management workflows. Addressing this capability gap demands a novel research agenda around AI-augmented lifecycle management tools.

The enterprise model deployment landscape is evolving rapidly, with a host of software vendors now offering MLOps tooling. But few enterprises are there yet—most are just beginning to deploy models into production environments and,

DOI: 10.17148/IJIREEICE.2022.101216

in many cases, are stalled at that phase due to a lack of supported tooling and processes. And among those enterprises that have begun to deploy models, most are still at an early state with limited tooling and processes to support any level of lifecycle management. Similarly, the tooling and capability landscape is evolving rapidly. Large, incumbent software vendors are introducing or growing MLOps tooling as are a host of start-ups. And model deployment, monitoring, and lifecycle management tooling is the fastest-growing segment of the modern software stack. However, this means that the landscape is highly fragmented and difficult to navigate.

Continuous research is essential to push the boundaries of the state of the art (especially at earlier stages of the development process where initial design and feasibility questions are addressed). This subjectivity (especially at earlier stages of the process) makes it difficult to obtain a comprehensive overview of MLOps and leads to individual interpretations across people and teams. MLOps is new enough that there is no comprehensive, technical literature or resources that have proven outcomes to support the relevant teams in designing and implementing sophisticated MLOps solutions. Such knowledge is essential, to shape the range of plausible options and evaluate candidates. Another limitation is that unknown unknowns preclude systematic consideration of alternatives beyond the inferred selection pool.

9.1. Future Trends

In the business domain, Enterprise Architecture (EA) has become critical for complex organizations as they have to deal with market changes at an unprecedented pace while providing an annual roadmap for the future. EA has ephemeral qualities that need to be taken into account. Information technology (IT) has been a crucial enabler of flexibility and agility within organizations. The pandemic crisis has accelerated certain already present trends regarding digital transformation and has disrupted traditional ways of working and communicating. Conventional organizations need to transform to agile organizations in order to maintain their competitive edge in the new reality. One of the options to transform is using the principles of collaborative processes to the agile development of products and services that realizes an executable architecture for the transformed organization. As agile transformation is not just a technology lift-and-shift, EA was expanded as a scholarly field to aid organizations in their transformation. All organizations require some form of management to survive. This has typically meant hierarchically structured chains of command to direct resources toward some goal. Yet, as organizations grow in size and complexity, numerous forms and arrangements of management are possible. Complex organizations operate with a mix of top-down planning and bottom-up emergent behavior grounded in shared vision, values, beliefs, and human cognition. A rudimentary model is proposed that captures the basic elements of this missing middle ground, unifying several theoretical perspectives on management. A mapping of desirable management forms to the model is presented to show its broader applicability. The ultimate goal is improved human understanding of the management variables, forms, and forces that affect all organizations. By modeling management at a new, higher level, the first essential step toward explaining why complex organizations behave the way they do has been taken. As a consequence, new capabilities to design, enact, and align communication, decision-making, and incentives are envisioned that will assist steering organizations toward their preferred emergent behavior.

Contributing to the explanation of the expected future trends of the growing MLOps domain, a series of questions are laid out. At the highest abstraction level, MLOps may become ever more ubiquitous due to the general rise in the use of machine learning (ML) across many industries. Organizations with more experience may see traditional software engineering (SE) and DevOps techniques being actuated in an ML context on a more granular level, leading to a more uniform landscape of applied methodologies and pragmatic tools. However, at a more granular level, there may also be quite unregulated capital-driven exuberance in the form of ML technology hype, with a concurrent zooming in on narrowly bounded vertical solutions that might be overseen and regulated at the data level but deploy AI solutions quite orthogonally with the data products.

As an alternative to these more conventional and additionally culturally dependent development scenarios, there is the view of ML through a sociotechnical lens. Key phenomena in this view revolve around investigating the emergence, societal entrenchment, and unintended consequences of ML, and AI in general. This necessarily includes populism, social inequality issues, and bias, but also more mundane societal issues such as working-from-home policies or even the rise in the rate of inflation. This rapid change in the assumed stable equilibrium may take more control methods, regulations, and governance mechanisms to properly unfold. In this view, the highly indeterminate outcome of the associated sociotechnical change process is in stark contrast to the somewhat deterministic goals of the SE and DevOps domains.

REFERENCES

[1] Kommaragiri, V. B., Preethish Nanan, B., Annapareddy, V. N., Gadi, A. L., & Kalisetty, S. (2022). Emerging Technologies in Smart Computing, Sustainable Energy, and Next-Generation Mobility: Enhancing Digital Infrastructure, Secure Networks, and Intelligent Manufacturing. Venkata Narasareddy and Gadi, Anil Lokesh and Kalisetty, Srinivas.

[2] Pamisetty, V., Dodda, A., Singireddy, J., & Challa, K. (2022). Optimizing Digital Finance and Regulatory Systems Through Intelligent Automation, Secure Data Architectures, and Advanced Analytical Technologies. Jeevani and Challa, Kishore, Optimizing Digital Finance and Regulatory Systems Through Intelligent Automation, Secure Data Architectures, and Advanced Analytical Technologies (December 10, 2022).

[3] Paleti, S. (2022). The Role of Artificial Intelligence in Strengthening Risk Compliance and Driving Financial Innovation in Banking. International Journal of Science and Research (IJSR), 11(12), 1424–1440. https://doi.org/10.21275/sr22123165037

[4] Komaragiri, V. B. (2022). Expanding Telecom Network Range using Intelligent Routing and Cloud-Enabled Infrastructure. International Journal of Scientific Research and Modern Technology, 120–137. https://doi.org/10.38124/ijsrmt.v1i12.490

[5] Pamisetty, A., Sriram, H. K., Malempati, M., Challa, S. R., & Mashetty, S. (2022). AI-Driven Optimization of Intelligent Supply Chains and Payment Systems: Enhancing Security, Tax Compliance, and Audit Efficiency in Financial Operations. Tax Compliance, and Audit Efficiency in Financial Operations (December 15, 2022).

[6] Mashetty, S. (2022). Innovations In Mortgage-Backed Security Analytics: A Patent-Based Technology Review. Kurdish Studies. https://doi.org/10.53555/ks.v10i2.3826

[7] Kurdish Studies. (n.d.). Green Publication. https://doi.org/10.53555/ks.v10i2.3785

[8] Motamary, S. (2022). Enabling Zero-Touch Operations in Telecom: The Convergence of Agentic AI and Advanced DevOps for OSS/BSS Ecosystems. Kurdish Studies. https://doi.org/10.53555/ks.v10i2.3833

[9] Kannan, S. (2022). AI-Powered Agricultural Equipment: Enhancing Precision Farming Through Big Data and Cloud Computing. Available at SSRN 5244931.

[10] Suura, S. R. (2022). Advancing Reproductive and Organ Health Management through cell-free DNA Testing and Machine Learning. International Journal of Scientific Research and Modern Technology, 43–58. https://doi.org/10.38124/ijsrmt.v1i12.454

[11] Nuka, S. T., Annapareddy, V. N., Koppolu, H. K. R., & Kannan, S. (2021). Advancements in Smart Medical and Industrial Devices: Enhancing Efficiency and Connectivity with High-Speed Telecom Networks. Open Journal of Medical Sciences, 1(1), 55-72.

[12] Meda, R. (2022). Integrating IoT and Big Data Analytics for Smart Paint Manufacturing Facilities. Kurdish Studies. https://doi.org/10.53555/ks.v10i2.3842

[13] Annapareddy, V. N., Preethish Nanan, B., Kommaragiri, V. B., Gadi, A. L., & Kalisetty, S. (2022). Emerging Technologies in Smart Computing, Sustainable Energy, and Next-Generation Mobility: Enhancing Digital Infrastructure, Secure Networks, and Intelligent Manufacturing. Venkata Bhardwaj and Gadi, Anil Lokesh and Kalisetty, Srinivas, Emerging Technologies in Smart Computing, Sustainable Energy, and Next-Generation Mobility: Enhancing Digital Infrastructure, Secure Networks, and Intelligent Manufacturing (December 15, 2022).

[14] Phanish Lakkarasu. (2022). AI-Driven Data Engineering: Automating Data Quality, Lineage, And Transformation In Cloud-Scale Platforms. Migration Letters, 19(S8), 2046–2068. Retrieved from https://migrationletters.com/index.php/ml/article/view/11875

[15] Kaulwar, P. K. (2022). Securing The Neural Ledger: Deep Learning Approaches For Fraud Detection And Data Integrity In Tax Advisory Systems. Migration Letters, 19, 1987-2008.

[16] Malempati, M. (2022). Transforming Payment Ecosystems Through The Synergy Of Artificial Intelligence, Big Data Technologies, And Predictive Financial Modeling. Big Data Technologies, And Predictive Financial Modeling (November 07, 2022).

[17] Recharla, M., & Chitta, S. (2022). Cloud-Based Data Integration and Machine Learning Applications in Biopharmaceutical Supply Chain Optimization.

[18] Lahari Pandiri. (2022). Advanced Umbrella Insurance Risk Aggregation Using Machine Learning. Migration Letters, 19(S8), 2069–2083. Retrieved from https://migrationletters.com/index.php/ml/article/view/11881

[19] Paleti, S., Burugulla, J. K. R., Pandiri, L., Pamisetty, V., & Challa, K. (2022). Optimizing Digital Payment Ecosystems: Ai-Enabled Risk Management, Regulatory Compliance, And Innovation In Financial Services. Regulatory Compliance, And Innovation In Financial Services (June 15, 2022).

[20] Singireddy, J. (2022). Leveraging Artificial Intelligence and Machine Learning for Enhancing Automated Financial Advisory Systems: A Study on AIDriven Personalized Financial Planning and Credit Monitoring. Mathematical Statistician and Engineering Applications, 71 (4), 16711–16728.

[21] Paleti, S., Singireddy, J., Dodda, A., Burugulla, J. K. R., & Challa, K. (2021). Innovative Financial Technologies: Strengthening Compliance, Secure Transactions, and Intelligent Advisory Systems Through AI-Driven Automation and Scalable Data Architectures. Secure Transactions, and Intelligent Advisory Systems Through AI-Driven Automation and Scalable Data Architectures (December 27, 2021).

[22] Sriram, H. K. (2022). Integrating generative AI into financial reporting systems for automated insights and decision support. Available at SSRN 5232395.

DOI: 10.17148/IJIREEICE.2022.101216

[23] Koppolu, H. K. R. (2021). Leveraging 5G Services for Next-Generation Telecom and Media Innovation. International Journal of Scientific Research and Modern Technology, 89–106. https://doi.org/10.38124/ijsrmt.v1i12.472

[24] End-to-End Traceability and Defect Prediction in Automotive Production Using Blockchain and Machine Learning. (2022). International Journal of Engineering and Computer Science, 11(12), 25711-25732. https://doi.org/10.18535/ijecs.v11i12.4746

[25] Chaitran Chakilam. (2022). AI-Driven Insights In Disease Prediction And Prevention: The Role Of Cloud Computing In Scalable Healthcare Delivery. Migration Letters, 19(S8), 2105–2123. Retrieved from https://migrationletters.com/index.php/ml/article/view/11883

[26] Sriram, H. K., ADUSUPALLI, B., & Malempati, M. (2021). Revolutionizing Risk Assessment and Financial Ecosystems with Smart Automation, Secure Digital Solutions, and Advanced Analytical Frameworks.

[27] Avinash Pamisetty. (2021). A comparative study of cloud platforms for scalable infrastructure in food distribution supply chains. Journal of International Crisis and Risk Communication Research , 68–86. Retrieved from https://jicrcr.com/index.php/jicrcr/article/view/2980

[28] Gadi, A. L., Kannan, S., Nanan, B. P., Komaragiri, V. B., & Singireddy, S. (2021). Advanced Computational Technologies in Vehicle Production, Digital Connectivity, and Sustainable Transportation: Innovations in Intelligent Systems, Eco-Friendly Manufacturing, and Financial Optimization. Universal Journal of Finance and Economics, 1(1), 87-100.

[29] Dodda, A. (2022). The Role of Generative AI in Enhancing Customer Experience and Risk Management in Credit Card Services. International Journal of Scientific Research and Modern Technology, 138–154. https://doi.org/10.38124/ijsrmt.v1i12.491

[30] Gadi, A. L. (2022). Connected Financial Services in the Automotive Industry: AI-Powered Risk Assessment and Fraud Prevention. Journal of International Crisis and Risk Communication Research, 11-28.

[31] Pamisetty, A. (2022). A Comparative Study of AWS, Azure, and GCP for Scalable Big Data Solutions in Wholesale Product Distribution. International Journal of Scientific Research and Modern Technology, 71–88. https://doi.org/10.38124/ijsrmt.v1i12.466

[32] Adusupalli, B. (2021). Multi-Agent Advisory Networks: Redefining Insurance Consulting with Collaborative Agentic AI Systems. Journal of International Crisis and Risk Communication Research, 45-67.

[33] Dwaraka Nath Kummari. (2022). Iot-Enabled Additive Manufacturing: Improving Prototyping Speed And Customization In The Automotive Sector . Migration Letters, 19(S8), 2084–2104. Retrieved from https://migrationletters.com/index.php/ml/article/view/11882

[34] Data-Driven Strategies for Optimizing Customer Journeys Across Telecom and Healthcare Industries. (2021). International Journal of Engineering and Computer Science, 10(12), 25552-25571. https://doi.org/10.18535/ijecs.v10i12.4662

[35] Adusupalli, B., Singireddy, S., Sriram, H. K., Kaulwar, P. K., & Malempati, M. (2021). Revolutionizing Risk Assessment and Financial Ecosystems with Smart Automation, Secure Digital Solutions, and Advanced Analytical Frameworks. Universal Journal of Finance and Economics, 1(1), 101-122.

[36] AI-Based Financial Advisory Systems: Revolutionizing Personalized Investment Strategies. (2021). International Journal of Engineering and Computer Science, 10(12). https://doi.org/10.18535/ijecs.v10i12.4655

[37] Karthik Chava. (2022). Harnessing Artificial Intelligence and Big Data for Transformative Healthcare Delivery. International Journal on Recent and Innovation Trends in Computing and Communication, 10(12), 502–520. Retrieved from https://ijritcc.org/index.php/ijritcc/article/view/11583

[38] Challa, K. (2022). The Future of Cashless Economies Through Big Data Analytics in Payment Systems. International Journal of Scientific Research and Modern Technology, 60–70. https://doi.org/10.38124/ijsrmt.v1i12.467

[39] Pamisetty, V., Pandiri, L., Annapareddy, V. N., & Sriram, H. K. (2022). Leveraging AI, Machine Learning, And Big Data For Enhancing Tax Compliance, Fraud Detection, And Predictive Analytics In Government Financial Management. Machine Learning, And Big Data For Enhancing Tax Compliance, Fraud Detection, And Predictive Analytics In Government Financial Management (June 15, 2022).

[40] Innovations in Spinal Muscular Atrophy: From Gene Therapy to Disease-Modifying Treatments. (2021). International Journal of Engineering and Computer Science, 10(12), 25531-25551. https://doi.org/10.18535/ijecs.v10i12.4659

[41] Kaulwar, P. K. (2022). Data-Engineered Intelligence: An AI-Driven Framework for Scalable and Compliant Tax Consulting Ecosystems. Kurdish Studies, 10 (2), 774–788.

[42] Operationalizing Intelligence: A Unified Approach to MLOps and Scalable AI Workflows in Hybrid Cloud Environments. (2022). International Journal of Engineering and Computer Science, 11(12), 25691-25710. https://doi.org/10.18535/ijecs.v11i12.4743

[43] Nandan, B. P., & Chitta, S. (2022). Advanced Optical Proximity Correction (OPC) Techniques in Computational Lithography: Addressing the Challenges of Pattern Fidelity and Edge Placement Error. Global Journal of Medical Case Reports, 2(1), 58-75.

[44] Raviteja Meda. (2021). Machine Learning-Based Color Recommendation Engines for Enhanced Customer Personalization. Journal of International Crisis and Risk Communication Research , 124–140. Retrieved from https://jicrcr.com/index.php/jicrcr/article/view/3018

[45] Rao Suura, S. (2021). Personalized Health Care Decisions Powered By Big Data And Generative Artificial Intelligence In Genomic Diagnostics. Journal of Survey in Fisheries Sciences. https://doi.org/10.53555/sfs.v7i3.3558

[46] Implementing Infrastructure-as-Code for Telecom Networks: Challenges and Best Practices for Scalable Service Orchestration. (2021). International Journal of Engineering and Computer Science, 10(12), 25631-25650. https://doi.org/10.18535/ijecs.v10i12.4671

[47] Vamsee Pamisetty, Lahari Pandiri, Sneha Singireddy, Venkata Narasareddy Annapareddy, Harish Kumar Sriram. (2022). Leveraging AI, Machine Learning, And Big Data For Enhancing Tax Compliance, Fraud Detection, And Predictive Analytics In Government Financial Management. Migration Letters, 19(S5), 1770–1784. Retrieved from https://migrationletters.com/index.php/ml/article/view/11808

[48] Someshwar Mashetty. (2020). Affordable Housing Through Smart Mortgage Financing: Technology, Analytics, And Innovation. International Journal on Recent and Innovation Trends in Computing and Communication, 8(12), 99–110. Retrieved from https://ijritcc.org/index.php/ijritcc/article/view/11581

[49] Srinivasa Rao Challa,. (2022). Cloud-Powered Financial Intelligence: Integrating AI and Big Data for Smarter Wealth Management Solutions. Mathematical Statistician and Engineering Applications, 71(4), 16842–16862. Retrieved from https://philstat.org/index.php/MSEA/article/view/2977

[50] Paleti, S. (2022). Fusion Bank: Integrating AI-Driven Financial Innovations with Risk-Aware Data Engineering in Modern Banking. Mathematical Statistician and Engineering Applications, 71(4), 16785-16800.

[51] Pamisetty, V. (2022). Transforming Fiscal Impact Analysis with AI, Big Data, and Cloud Computing: A Framework for Modern Public Sector Finance. Big Data, and Cloud Computing: A Framework for Modern Public Sector Finance (November 30, 2022).

[52] Kommaragiri, V. B., Gadi, A. L., Kannan, S., & Preethish Nanan, B. (2021). Advanced Computational Technologies in Vehicle Production, Digital Connectivity, and Sustainable Transportation: Innovations in Intelligent Systems, Eco-Friendly Manufacturing, and Financial Optimization.

[53] Annapareddy, V. N. (2022). Integrating AI, Machine Learning, and Cloud Computing to Drive Innovation in Renewable Energy Systems and Education Technology Solutions. Available at SSRN 5240116.

[54] Transforming Renewable Energy and Educational Technologies Through AI, Machine Learning, Big Data Analytics, and Cloud-Based IT Integrations. (2021). International Journal of Engineering and Computer Science, 10(12), 25572-25585. https://doi.org/10.18535/ijecs.v10i12.4665

[55] Venkata Bhardwaj Komaragiri. (2021). Machine Learning Models for Predictive Maintenance and Performance Optimization in Telecom Infrastructure. Journal of International Crisis and Risk Communication Research , 141–167. Retrieved from https://jicrcr.com/index.php/jicrcr/article/view/3019

[56] Paleti, S. (2021). Cognitive Core Banking: A Data-Engineered, AI-Infused Architecture for Proactive Risk Compliance Management. AI-Infused Architecture for Proactive Risk Compliance Management (December 21, 2021).

[57] Harish Kumar Sriram. (2022). AI-Driven Optimization of Intelligent Supply Chains and Payment Systems: Enhancing Security, Tax Compliance, and Audit Efficiency in Financial Operations. Mathematical Statistician and Engineering Applications, 71(4), 16729–16748. Retrieved from https://philstat.org/index.php/MSEA/article/view/2966

[58] Chava, K., Chakilam, C., Suura, S. R., & Recharla, M. (2021). Advancing Healthcare Innovation in 2021: Integrating AI, Digital Health Technologies, and Precision Medicine for Improved Patient Outcomes. Global Journal of Medical Case Reports, 1(1), 29-41.

[59] Data Engineering Architectures for Real-Time Quality Monitoring in Paint Production Lines. (2020). International Journal of Engineering and Computer Science, 9(12), 25289-25303. https://doi.org/10.18535/ijecs.v9i12.4587

[60] Pallav Kumar Kaulwar. (2021). From Code to Counsel: Deep Learning and Data Engineering Synergy for Intelligent Tax Strategy Generation. Journal of International Crisis and Risk Communication Research , 1–20. Retrieved from https://jicrcr.com/index.php/jicrcr/article/view/2967

[61] Pandiri, L., & Chitta, S. (2022). Leveraging AI and Big Data for Real-Time Risk Profiling and Claims Processing: A Case Study on Usage-Based Auto Insurance. Kurdish Studies. https://doi.org/10.53555/ks.v10i2.3760

[62] Kummari, D. N. (2022). AI-Driven Predictive Maintenance for Industrial Robots in Automotive Manufacturing: A Case Study. International Journal of Scientific Research and Modern Technology, 107–119. https://doi.org/10.38124/ijsrmt.v1i12.489

DOI: 10.17148/IJIREEICE.2022.101216

[63] Gadi, A. L. (2022). Cloud-Native Data Governance for Next-Generation Automotive Manufacturing: Securing, Managing, and Optimizing Big Data in AI-Driven Production Systems. Kurdish Studies. https://doi.org/10.53555/ks.v10i2.3758

[64] Dodda, A. (2022). Secure and Ethical Deployment of AI in Digital Payments: A Framework for the Future of Fintech. Kurdish Studies. https://doi.org/10.53555/ks.v10i2.3834

[65] Gadi, A. L. (2021). The Future of Automotive Mobility: Integrating Cloud-Based Connected Services for Sustainable and Autonomous Transportation. International Journal on Recent and Innovation Trends in Computing and Communication, 9(12), 179-187.

[66] Dodda, A. (2022). Strategic Financial Intelligence: Using Machine Learning to Inform Partnership Driven Growth in Global Payment Networks. International Journal of Scientific Research and Modern Technology, 1(12), 10-25.

[67] Just-in-Time Inventory Management Using Reinforcement Learning in Automotive Supply Chains. (2021). International Journal of Engineering and Computer Science, 10(12), 25586-25605. https://doi.org/10.18535/ijecs.v10i12.4666

[68] Srinivasa Rao Challa. (2021). From Data to Decisions: Leveraging Machine Learning and Cloud Computing in Modern Wealth Management. Journal of International Crisis and Risk Communication Research , 102–123. Retrieved from https://jicrcr.com/index.php/jicrcr/article/view/3017

[69] Kommaragiri, V. B. (2021). Enhancing Telecom Security Through Big Data Analytics and Cloud-Based Threat Intelligence. Available at SSRN 5240140.