DevOps for Data Science by Bridging the Gap between Development and Data Pipelines

Sumanth Tatineni

Abstract: ***The evolving technology landscape requires the convergence of DevOps and Data Science, which has become a pivotal force by combining innovation and efficiency to empower organizations with data-driven insights. While traditionally related to software development, DevOps has increased its influence on data science, creating a mutual relationship that bridges the gap between data analysis and development. This article explores the huge significance of implementing DevOps practices in the data science industry, thus addressing challenges and displaying the transformative benefits for organizations aiming to utilize the full potential of their data assets. The collaboration between DevOps and Data Science may initially seem like an unlikely pairing, particularly given their distinct focuses on software development and data analysis. However, this merger holds a huge promise for organizations looking to maximize the value extracted from their data. This paper delves into the DevOps for Data Science concept, depicting how this collaboration accelerates decision-making processes by promoting faster, more reliable and insightful outcomes. The intersection of DevOps and Data Science regarding data-driven decision-making is important for business success. This article explores how DevOps integrates into data science with its core principles of collaboration, automation, and continuous improvement by addressing challenges related to the traditional division between development and data analytics. DevOps practices revolutionize how organizations extract from theory data, mainly reshaping the decision-making approach. The article emphasizes the practical application of DevOps in data science and its role in transforming the reliability and efficiency of overall development. DevOps is slowly gaining recognition as a strong solution for breaking down traditional barriers between operations and developers in contemporary organizations. By emphasizing efficient teamwork and automation, the article highlights how DevOps accelerates delivery speed, promoting overall organizational performance and providing a competitive edge in the market.***

Keywords: DevOps, Data Science, integration, machine learning models, automation, data analytics, and continuous integration

1. Introduction

Integrating DevOps in data science shows a huge shift in how organizations utilize data's power to enhance business value. Siloed operations are from the past, especially as DevOps practices systematically dismantle barriers that traditionally separated development and data analytics [1]. This collaboration allows a seamless flow of information, allowing organizations to derive faster, more precise insights that directly affect decision-making processes.

DevOps acts as a block that brings together the different strengths of development and data science teams, thus promoting collaboration beyond the limitations of conventional structures [2]. The result is the merging of skills and expertise that expedites the pace of innovation and pushes organizations toward success. As modern organizations strove to extract helpful insights from different data points, the collaboration between DevOps and data science is the solution for purposeful and dynamic partnerships. This shift positions organizations on the lead of industry advancements.

2. Significance and Background of DevOps for Data Science

In traditional development environments, merging data science and conventional development processes brings different challenges. The innate disconnect between the objectives, methodologies, and tools applied by data scientists and their counterparts in traditional development teams pose some obstacles. While data scientists are immersed in data exploration, model building, and insight extraction, traditional development teams focus on delivering software products. The differences often result in the slow deployment of data-driven solutions and hinder collaborative efforts between these two domains.

The unique methodologies applied in data science, such as machine learning, statistical analysis, and data exploration, may not seamlessly align with the established processes of traditional software development [3]. This misalignment depicts the critical need for bridging the gap between data science and development for organizations aspiring to harness data-driven potency for decision-making and secure a competitive scope in today's data-centric scope. Addressing this disparity is key. DevOps principles and practices are a solution to surmount these challenges and enhance the efficiency of data science projects.

Embracing DevOps methodologies allows data science teams to seamlessly integrate into the development pipeline, thus aligning their workflows with the development and operations complexities. Collaboration, automation, and continuous integration set organizations up for success by empowering data scientists to deliver helpful insights and predictive models quickly and accurately. The strategic integration of DevOps principles bridges the historical gap, thus promoting a coordinated coexistence between data science and development for organizations poised to thrive in an era driven by data [4].

3. DevOps Principles for Data Science

Implementing DevOps principles in data science is a strategic shift that combines collaborative practices and efficient methodologies. This approach aims to streamline the processes of data processing, modeling, and deployment, which effectively minimizes bottlenecks in deployment and ensures a high degree of consistency across the entire data pipeline. The implementation of DevOps for data science is interchangeable with accelerated time-to-insights [5], increased reliability through automated testing, and the cultivation of cross-functional collaboration. Despite the challenge of developing from the different nature of data and the requirement for specialized tools, there are compelling advantages of achieving faster and more accurate insights. Here are the core principles and how they can be applied to the data science landscape;

3.1. Collaboration

Collaboration fosters an environment where cross-functional teams seamlessly unite to achieve common goals. This collaboration is key for software development, which mirrors the need within the data science domain; the integration of these collaborative principles between data science, development, and domain expertise is advantageous and is key for the success of modern organizations. In data science, bringing together different perspectives of data scientists, developers, and domain experts is key [6]. It forms an alliance where every team's unique skills and insights contribute to a more comprehensive understanding of business objectives. Their expertise in data exploration and statistical analysis allows data scientists to collaborate closely with developers who excel in building robust and scalable systems. Including domain experts ensures that the data-driven insights derived from the analysis are accurate and aligned with the organization's main goals and vision. The collaboration ensures that data science projects are not pursued in isolation but align with the organization's strategic objectives.

3.2. Version control and reproducibility

Version control systematically tracks code, data, and model changes throughout the development and analysis processes. The tracking serves two purposes: it preserves a detailed historical record of the evolution of the data science project, thus allowing teams to revisit and understand the progression of analyses over time. Secondly, it ensures the reproducibility of analyses by providing an idea of the exact state of code and data at any given point in the project's timeline[7]. In data science, where analyses often include complex data manipulations, fragile modeling techniques, and different algorithms, version control becomes a safeguard against uncertainty and ambiguity.

Using well-established tools like Git, which has been key in software development, and platforms like DVC, particularly tailored for managing the versioning of data science projects, ensures that the version control process seamlessly aligns with the unique needs of data scientists. By adopting these version control tools, data science teams ensure the reproducibility of their analyses and cultivate an environment conducive to collaborative workflows. Every iteration, improvement, or adjustment to the models or code becomes a trackable and transparent event. This transparency facilitates better collaboration within the data science teams and enables cross-functional cooperation with development and operation teams with the DevOps framework.

3.3. Automation

Data processing, model training, and deployment are key components of the analytical journey in data science for efficiency and accuracy. By automating these repetitive tasks, data scientists can liberate valuable time and cognitive resources from mundane activities, thus redirecting their focus toward higher-order thinking and more strategic aspects of their analyses. One of the key advantages of automation in data science is the mitigation of manual errors. When tackled manually, repetitive tasks are prone to human errors that can greatly affect the accuracy of analyses.

Automation becomes a safeguard against such errors, thus ensuring that every step of the data science process is executed reliably and consistently [8], which enhances the precision of analyses and contributes to the creation of more reproducible and trustworthy results. The acceleration of analysis is a key benefit brought about by automating data science workflows. Complex tasks like data processing, which traditionally require a significant investment of effort and time, can be streamlined and executed swiftly through automated processes. Model training can also be optimized for efficiency since it isa resource-intensive phase that enables data scientists to experiment with various models in a more iterative and agile.

3.4. Continuous integration and continuous deployment

CI/CD is a set of best practices to ensure software's rapid and dependable release. When this principle gets applied to data science, it brings about a shift wherein the iterative nature of modeling and analysis is seamlessly merged into a cohesive and automated workflow. Adopting CI/CD practices in data science pushes teams to engage in swift iterations, thus allowing for dynamic adjustments in modeling and analysis in response to evolving data or emerging insights. This approach, akin to the frequent software releases in traditional DevOps settings, thus facilitates the seamless evolution of data science projects. As data evolves or models are refined, the CI/CD pipeline ensures that these changes are rigorously tested through automated testing and validation pipelines before deployment.

Automated testing in this process systematically validates each modification to the models or data to ensure that the integrity of the analysis is maintained. This guarantees the reliability of the data-driven decision-making process and instills confidence in the outcomes, knowing adjustments have been thoroughly scrutinized before integration into the production environment. In addition, the CI/CD pipeline in data science ensures consistency and repeatability, which is key in promoting trust in decision-making [9]. Every workflow step, from data ingestion to model deployment, is su, object to automated validation, thus minimizing the risk of introducing discrepancies and errors.

3.5. Infrastructure as Code (IaC)

IaC involves codifying infrastructure provisioning and treating infrastructure configurations as code artifacts.

Extending these principles to data sciences is a strategic step that deals with the inherent challenges of creating and maintaining consistent environments for analysis across different stages of production and development. Adopting containerization and orchestration tools like Kubernetes and Docker further improves the application of IaC principles in data science. Containers incorporate the entire environment's required data analysis into one package, like dependencies, libraries, and runtime.

This compartmentalization ensures that the analysis environment remains consistent and can be effortlessly replicated across different computing environments [10]. In addition, orchestration tools like Kubernetes provide an efficient and scalable means of managing and deploying these containerized environments. This ensures that the uniformity achieved during development is seamlessly extended into the production stage. The ability to deploy, scale, and consistently manage data science workloads becomes a key advantage, promoting reliability and predictability in the deployment process.

3.6. Monitoring and feedback loops

This principle reflects a strategic alignment highlighting the importance of continuous adaptability and oversight in analytical model development. Drawing parallels with the vigilance used to applications in production within the DevOps space, the application of monitoring and feedback loops in data science becomes a key practice for ensuring the ongoing relevance, accuracy, and efficacy of analytical models. Models are instrumental in data science in extracting insights from data; the concept of continuous monitoring shows the key approach adopted in DevOps for applications in production.

This incorporates the systematic observation and measurement of key metrics associated with model performance, data distributions, and other relevant indicators [11]. Continuous monitoring assists in detecting any deviations or shifts in these metrics, which could affect the relevance and accuracy of the models. The significance of this principle is that the mechanisms enable the timely assimilation of insights gained from monitoring into the repetitive development process of data science models.

When data distribution shifts or model performance changes are detected, feedback loops trigger a response often involving updates, recalibration, or even model retraining. The repetitive feedback mechanism ensures that models remain adaptive to evolving conditions and continue to deliver accurate and relevant results over time. One of the main advantages of implementing monitoring and feedback loops in data science is the ability to address issues promptly before they escalate.

By continuously assessing the performance and relevance of models, organizations can preemptively identify and rectify any anomalies or deviations, thus ensuring data-driven decisions are based on the most accurate and up-to-date insights [12]. In addition, this repetitive nature seamlessly aligns with the development practices often employed in data science. It creates a cycle that is not static but evolves in response to changing data patterns and emerging trends.

4. Benefits of DevOps in Data Science

4.1. Faster time-to-insights

DevOps practices can expedite data science projects by dismantling barriers, optimizing workflows, and promoting a culture of continuous improvement. This streamlined technique reduces bottlenecks, allowing quicker model development and deployment iteration. This, in return, leads to reduced time-to-insights, thus allowing organizations to gather valuable information from their data more timely.

4.2. Improved collaboration

As mentioned, DevOps principles provide easy collaboration, and this ethos is seamlessly translated into the data science space. A shared understanding of project goals is nurtured by encouraging communication between development and data science teams. This collaboration breaks down silos and also facilitates the creation of automated and reproducible data processing, modeling, and analysis pipelines, thus leading to more efficient and faster insights.

4.3. Reduced risks

One of the key benefits DevOps brings to data science is risk reduction. CI/CD practices minimize the risks of deploying faulty models or introducing errors into production systems. Stringent automated testing ensures that insights derived from data are accurate and reliable. Risk mitigation is key to maintaining the integrity of the decision-making process.

4.4. Scalability

Using DevOps practices in data science ensures scalability, which is crucial, especially with the ever-growing data volumes. Data pipelines and analysis workflows become capable of handling larger and more complex datasets. This scalability future-proofs data science projects and initiatives and positions organizations to leverage the full potential of the data assets.

4.5. Improved integration for maximum value extraction

The merge between Data science and DevOps allows organizations to discover a potent formula for maximizing the value derived from the data assets. While challenges like navigating the difficulties of model deployment and cultural shifts may exist, aligning these two disciplines empowers organizations to stay at the front of innovation in the data-driven eta. This holistic integration of DevOps principles ensures efficiency and reliability and positions organizations to extract the maximum value from their data, thus driving success and innovation in this dynamic landscape.

5. Challenges and considerations

Navigating the integration of DevOps with data science brings about different benefits and equally a lot of challenges and considerations. The difficulties of dealing with different and complex data and the need for specialized tools require proper planning [13]. This integration often requires adjustments to established workflows and innovative technologies that cater to the distinctive requirements of both development and data domains. In addition, addressing cultural shifts within teams is a vital step for successfully implementing DevOps for data science.

Achieving collaboration needs promoting open communication, thus nurturing mutual understanding and cultivating the willingness to bridge the gaps between traditionally distinct roles. Getting to this collaboration depicts the importance of acknowledging and addressing these challenges; one huge problem is overcoming the cultural shift relevant to dismantling the traditional separation between development and data science, requiring a concerted effort to align goals and promote a collaboration culture.

In addition, the need for data governance is huge, particularly concerning data privacy and compliance when shared across teams. Implementing stringent data governance practices is important to mitigate risks [14]. Consequently, adopting DevOps in data science brings about considerations related to tooling and skills. Data scientists may find themselves adapting to new tools and practices while developers may be required to acquaint themselves with fundamental data science concepts; thus, bridging the knowledge gap is important to create a harmonious collaboration between these domains.

6. Enhancing collaboration with DevOps

DevOps incorporates different tasks like configuration management, continuous integration, deployment, testing, monitoring, and infrastructure provisioning. Traditionally, DevOps teams closely worked with development teams to manage the lifecycle of applications effectively. However, the integration of data science in DevOps brings forth new responsibilities. Data engineering, for instance, is a specialized category that deals with data pipelines, which needs close cooperation between DevOps and data science teams.

DevOps operators are tasked with provisioning strong clusters for technologies such as Apache Airflow or Apache Hadoop, which are key for data transformation and extraction [15]. Data engineers navigate different data sources, leveraging Big Data clusters and pipelines to transform raw data. Subsequently, data scientists use tools such as Power BI and Notebooks to visualize and explore transformed data. DevOps teams are essential in providing environments tailored for data visualization and exploration.

Building machine learning models brings about a huge shift from traditional application development. It is a heterogeneous process that involves different languages, toolkits, libraries, and development environments like Python or Visual Studio Code [16]. DevOps teams must ensure these environments are readily available for data scientists and developers working on ML issues. The resource-intensive nature of machine learning and deep learning needs substantial computing infrastructure with powerful CPUs and GPUs. DevOps handles tasks like provisioning, configuring, scaling, and managing clusters for frameworks like Microsoft CNTK and TensorFlow.

Automation is key in the DevOps toolkit as it is used to streamline these processes, including creating and terminating instances. ML development is key to employing CI/CD best practices like modern application development. Every version of an ML model is packaged as a container image, tagged differently, and managed through sophisticated CI/CD pipelines [17]. When a fully-trained ML model is ready, DevOps teams are responsible for hosting it in a scalable environment, often leveraging orchestration engines like Kubernetes.

Containers and container management tools have significantly eased ML development, providing manageability and efficiency. DevOps teams embrace containers for different reasons, such as provisioning development environments, data processing pipelines, training infrastructure, and model deployment environments. Emerging technologies such as Kube Flow are specifically designed to empower DevOps teams in navigating the unique challenges posed by ML infrastructure [18]. ML brings a new path to DevOps, requiring collaborative efforts between operators, data scientists, developers, and data engineers to support businesses embracing the ML pattern effectively.

7. Versioning and rollback

The significance of versioning and rollback in the lifecycle management of deployed ML models is key in tracking the evolution of the models over time, capturing changes and improvements, and allowing for comprehensive comparisons between various versions. This includes assessing outputs, inputs, parameters, metrics, and code. On the other hand, rollback functionality is important in ensuring swift restoration to a prior model version in case of errors, bugs, or performance deterioration. This combination ensures the ML model's maintenance of quality, consistency, and reproducibility [19]. In addition, they empower organizations to adapt to changing data, evolving requirements, and valuable feedback, thus promoting agility in ML model management.

Organizations can systematically captures and stores all dependencies and components constituting the model, data, code, configuration, environment, artifacts, and metadata to achieve this. Every model version needs a distinct identifier, timestamp, and comprehensive documentation outlining the changes and reasoning behind them. Tools such as MLflow can significantly help in the versioning process. These tools integrate with existing data sources, code repositories, and deployment platforms, thus presenting one interaction that efficiently manages model versions.

This contributes to the efficient development of model versions, thus emphasizing the merger between development and data science teams [20]. Implementing a rollback strategy for your ML model should entail establishing a mechanism that allows seamless switching between different model versions within the organization's production environment without compromising service performance and availability. Employing a deployment strategy that fully supports rollback is essential to achieve this.

An example is the rolling deployment strategy, where updates or new versions of the ML model are introduced across nodes or instances in the production environment. This incremental update allows for a gradual traffic transition from instances running the old version to those with the new version, thus ensuring a smooth deployment process [21]. Throughout this transition, key performance indicators are monitored carefully to verify that the new version operates as expected.

Most importantly, the rolling deployment strategy provides the flexibility to roll back or halt the deployment if any issues are detected, or the new version fails to meet performance criteria. After all is done, it is important to validate and enhance the effectiveness of your versioning and rollback processes. Rigorous testing of your ML model is important, particularly from pre-deployment, deployment, and post-deployment. The testing protocol comprehensively examines the ML model's functionality, quality, and compatibility with key components such as environment, service, code, and data.

Most importantly, testing evaluates the impact and outcomes of versioning and rollback actions, such as performance metrics, stability, and user satisfaction within the model service [22]. This testing is a proactive measure that helps prevent and detect errors, bugs, and potential failures. It ensures the ML model aligns with the predefined requirements and meets the stakeholders' expectations. The framework is strengthened by integrating strong testing practices [23], thus fostering a collaborative environment that bridges the gap between development and data pipelines, highlighting reliability and quality assurance.

8. The future

The merge of DevOps and data science, is poised to experience evolution with different key trends. Different tools designed for the fusion of these two are expected to streamline processes and promote enhanced collaboration. As the collaboration advances, automation will expand its abilities across the whole data science workflow. This will include the model deployment and development automation and the orchestration of end-to-end data science processes. Advanced automation tools will streamline repetitive tasks, thus allowing teams to focus on higher-value activities [24], thus enhancing efficiency and reducing the need for manual intervention.

In addition, the adoption of containerization technologies and microservices architecture will be more widespread. This will allow organizations to encapsulate data science applications and their dependencies within portable containers, thus facilitating scalable and consistent deployment across different environments. Subsequently, embracing these architectures will enable a modular and flexible approach to building and maintaining data science solutions.

Consequently, MLOps, an extension of DevOps curated for ML, will grow to address the unique challenges of managing the entire ML model lifecycle. Organizations will increasingly focus on establishing strong practices for versioning, testing, deploying, monitoring, and maintaining ML models [26]. This will ensure that machine learning seamlessly integrates into the broader DevOps, thus promoting a more efficient and holistic data-driven decision-making process.

The future will have specialized tools for DevOps in data science, catering to the unique needs of integrating data science workflows into DevOps pipelines, thus providing organizations with dedicated resources to navigate the difficulties of collaborative model deployments and development [27]. Artificial intelligence for operations (AIOps) will revolutionize the management of DevOps pipelines and processes. AI and ML algorithms will optimize decision-making processes, automate routine tasks, and provide predictive insights. This merge will enhance the efficiency and adaptability of DevOps practices, thus making them more responsive to dynamic operational needs.

9. Conclusion

Organizations adopting the intersection between DevOps are set to be propelled to the front in this data-driven era. Bridging the traditional gap between development and data analytics allows teams to collaborate seamlessly [28], thus getting the true potential of data-driven insights. This merge accelerates time-to-insights and ensures their reliability, precision, and a huge impact on decision-making. DevOps for data science is collaboration and a transformative move for revolutionizing how organizations extract business value from their data [29]. The approach to building, deploying, and maintaining data-driven applications brings a new era of efficiency and innovation.

As organizations increasingly depend on data-driven decision-making, embracing DevOps is key to success in the competitive data-driven space. DevOps is a methodology that will continue to reshape the software development and deployment space. This shift emphasizing collaboration, automation, and continuous improvement allows organizations to deliver high-quality applications with reliability and speed. By bridging the gap between these two, DevOps teams will lean more towards shared goals [30], thus ensuring exceptional experiences and adaptability in case of any evolving demands of the digital world. Organizations embracing this collaboration will be equipped to navigate the challenges and opportunities of the data-driven future with the right innovative strategies.

References

[1] Ereth, J. (2018). DataOps-Towards a Definition. *LWDA*, *2191*, 104-112.

[2] Lwakatare, L. E., Kuvaja, P., &Oivo, M. (2016). An exploratory study of devops extending the dimensions of devops with practices. *Icsea*, *104*, 2016.

[3] Grady, N. W., Payne, J. A., & Parker, H. (2017, December). Agile big data analytics: AnalyticsOps for data science. In *2017 IEEE international conference on big data (big data)* (pp. 2331-2339). IEEE.

[4] Bou Ghantous, G., & Gill, A. (2017). DevOps: Concepts, practices, tools, benefits and challenges. *PACIS2017*.

[5] Ivanova, A., & Ivanova, P. (2018). DATA ANALYTICS FOR DEVOPS EFFECTIVENESS.

[6] Fokaefs, M., Barna, C., Veleda, R., Litoiu, M., Wigglesworth, J., & Mateescu, R. (2016, October). Enabling devops for containerized data-intensive applications: an exploratory study. In *Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering* (pp. 138-148).

[7] Muñoz, M., & Díaz, O. (2017). DevOps: Foundations and its utilization in data center. *Engineering and Management of Data Centers: An IT Service Management Approach*, 205-225.

[8] Lwakatare, L. E., Kuvaja, P., &Oivo, M. (2016). Relationship of devops to agile, lean and continuous deployment: A multivocal literature review study. In *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17* (pp. 399-415). Springer International Publishing.

[9] Angara, J., Prasad, S., & Sridevi, G. (2017). The factors driving testing in devops setting-a systematic literature survey. *Indian Journal of Science and Technology*, *9*(48), 1-8.

[10] George, G., Osinga, E. C., Lavie, D., & Scott, B. A. (2016). Big data and data science methods for management research. *Academy of Management Journal*, *59*(5), 1493-1507.

[11] Cao, L. (2017). Data science: a comprehensive overview. *ACM Computing Surveys (CSUR)*, *50*(3), 1-42.

[12] Vartak, M., & Madden, S. (2018). Modeldb: Opportunities and challenges in managing machine learning models. *IEEE Data Eng. Bull.*, *41*(4), 16-25.

[13] Baylor, D., Breck, E., Cheng, H. T., Fiedel, N., Foo, C. Y., Haque, Z., ... & Zinkevich, M. (2017, August). Tfx: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1387-1395).

[14] Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., ... & Zumar, C. (2018). Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.*, *41*(4), 39-45.

[15] Wu, D., Zhu, L., Xu, X., Sakr, S., Sun, D., & Lu, Q. (2016). Building pipelines for heterogeneous execution environments for big data processing. *IEEE Software*, *33*(2), 60-67.

[16] Wang, R., Sun, D., Li, G., Atif, M., & Nepal, S. (2016, December). Logprov: Logging events as provenance of big data analytics pipelines with trustworthiness. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 1402-1411). IEEE.

[17] Steele, B., Chandler, J., & Reddy, S. (2016). *Algorithms for data science* (pp. 1-430). New York: Springer.

[18] Kotu, V., & Deshpande, B. (2018). *Data science: concepts and practice*. Morgan Kaufmann.

[19] McMaster, K., Wolthuis, S. L., Rague, B., & Sambasivam, S. (2018). A comparison of key concepts in data analytics and data science. *Information Systems Education Journal*, *16*(1), 33.

[20] Draxl, C., & Scheffler, M. (2018). NOMAD: The FAIR concept for big data-driven materials science. *Mrs Bulletin*, *43*(9), 676-682.

[21] Liu, S., Wang, X., Liu, M., & Zhu, J. (2017). Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, *1*(1), 48-56.

[22] Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

[23] Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., ... & Song, D. (2017). Robust physical-world attacks on machine learning models. *arXiv preprint arXiv:1707.08945*, *2*(3), 4.

[24] Suthaharan, S. (2016). Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst*, *36*, 1-12.

[25] Kim, B. (2015). *Interactive and interpretable machine learning models for human machine collaboration* (Doctoral dissertation, Massachusetts Institute of Technology).

[26] Kamuto, M. B., & Langerman, J. J. (2017, May). Factors inhibiting the adoption of DevOps in large organisations: South African context. In *2017 2nd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)* (pp. 48-51). IEEE.

[27] Donoho, D. (2017). 50 years of data science. *Journal of Computational and Graphical Statistics*, *26*(4), 745-766.

[28] Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., & Yaqoob, I. (2017). Big IoT data analytics: architecture, opportunities, and open research challenges. *ieee access*, *5*, 5247-5261.

[29] Mottin, D., Lissandrini, M., Velegrakis, Y., &Palpanas, T. (2017). New trends on exploratory methods for data analytics. *Proceedings of the VLDB Endowment*, *10*(12), 1977-1980.

[30] Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., ... & Kumar, V. (2017). Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, *29*(10), 2318-2331.