2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) | 978-1-7281-7628-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/SYNASC51798.2020.00015

Sustainable MLOps: Trends and Challenges

Damian A. Tamburri
Eindhoven University of Technology - JADS
's-Hertogenbosch, The Netherlands
d.a.tamburri@tue.nl

***Abstract*—Even simply through a GoogleTrends search it becomes clear that Machine-Learning Operations—or *MLOps*, for short—are climbing in interest from both a scientific and practical perspective. On the one hand, software components and middleware are proliferating to support all manners of MLOps, from AutoML (i.e., software which enables developers with limited machine-learning expertise to train high-quality models specific to their domain or data) to feature-specific ML engineering, e.g., Explainability and Interpretability. On the other hand, the more these platforms penetrate the day-to-day activities of software operations, the more the risk for AI Software becoming unsustainable from a social, technical, or organisational perspective. This paper offers a concise definition of MLOps and AI Software Sustainability and outlines key challenges in its pursuit.**

***Keywords*-Machine-Learning Operations, MLOps, DataOps, Software Sustainability**

I. INTRODUCTION

Google trends puts Machine-Learning Operations as one of the most promisingly increasing trends (see Fig. 1). This paper looks at defining Machine-Learning Operations' sustainability and draws a research roadmap in its pursuit. First, from an operational perspective, a Machine-Learning software itself behaves no differently than any other software: it receives a properly-formatted input and produces an output [1].

Conversely, when it comes to operating such a Machine-Learning software continuously, i.e., as part of a DataOps pipeline, things change radically [2]. In essence, Machine-Learning Operations—or *MLOps*—entail the complex cloud orchestration [3] of a set of middleware and software components together realising at least 5 functions: (1) data ingestion/transport; (2) data transformation; (3) continuous ML (re-)training; (4) continuous ML (re-)deployment; (5) output production/presentation to the end-user (e.g., in the form of business intelligence [4]). All aforementioned functions reflect a diverse set of software components to be orchestrated as part of the entire solution—which is often referred to as *AI Software* [5]—for the purpose of realising its intended business function. It should also be noted that where MLOps meets Cloud engineering, there exist several technical devices and design patterns that enable such MLOps to become even more efficient—e.g., consider the adoption of Function-as-a-Service [6] within data pipelines—and effective from an

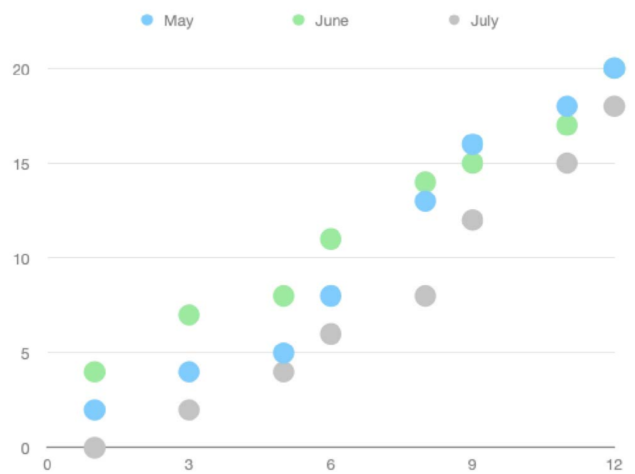

Figure 1. Trends for MLOps searches online for year 2020 alone; a variant super-linear growth is evident.

operational perspective but definitely more complex from an architectural perspective.

On the one hand, both research and practice have concentrated in producing a proliferation of tools and components to support the definition and improved operation of such AI software. Consider for example, software such as Apache AirFlow[1] or KubeFlow[2], which support the lifecycle operations of ML components within an AI Software. Similarly, consider software such as Google Cloud AutoML[3] which enables, in principle, any end-user to quickly put together an ML autonomous decision mechanism while hiding both the complexity of ML modelling exercises [7] and the lifecycle operations behind them.

On the other hand, the more complexity is added to AI software operations, the less such operations become *sustainable* [8]. The term *complexity* is intended not only software architecture complexity (e.g., even simply the number of components to be orchestrated as part of the AI software solution) but also the orchestration management complexity intrinsic to handling many autonomous software components at the same time and continuously [9]. Similarly, the term

[1]https://airflow.apache.org/
[2]https://www.kubeflow.org/
[3]https://cloud.google.com/automl/docs

978-1-7281-7628-4/20/$31.00 ©2020 IEEE
DOI 10.1109/SYNASC51798.2020.00015

sustainability is intended as the ability of the AI software itself to exist and operate continuously while keeping at least three operational objectives addressed: (1) from a technical perspective, the software shall maintain itself operational; (2) from an organisational perspective, the solution shall enable its own operations to be improved by the organisational structure around it with minimal efforts; (3) the solution shall continuously improve its intended social function without counteracting towards its own intended social contract [10].

The rest of this paper offers a deeper overview of the aforementioned concepts, starting from background and related work into the matter (see Sec. II). Subsequently, the paper outlines the challenges currently envisioned at an educational level around AI software (see Sec. III). Finally, the paper concludes with final remarks (Sec. V).

II. Background and Related Work

This section outlines fundamental notions around previously introduced matters and showcases a brief overview into the state-of-the-art.

A. Software Sustainability: A Primer

Software sustainability is a multi-faceted concept which draws from an equally diverse state-of-the-art. Seacord et. al.,[11] view sustainability in relation to "all activities related to software evolution and the ability to modify a software system based on stakeholders changing requirements". The aforementioned definition focuses on software maintenance activities as well as the matching between such activities and the human stakeholders around them.

Conversely, a more varied definition is brought about by Calero, Bertoa and Moraga [12] who define sustainable software as "a mode of software development in which resource use aims to meet product software needs while ensuring the sustainability of natural systems and the environment" and therefore focus on the matching between the software and how its operational footprint influences the forces and natural systems from whence it draws.

Similarly, Razavian et al. [8] elaborate on sustainable from a software services perspective, offering an overview of 4 key facets from which software sustainability must be defined, namely, *"We need e-services that are economically, technically, environmentally, and socially sustainable: economic sustainability to ensure that e-services create economic value; technical sustainability so that their technical assets actually enable the e-services to cope with changes; environmental sustainability to avoid that e-services harm the environment they operate in; social sustainability to ensure e-services provide fair exchange of information between parties"*.

Although several attempts have been made to define software sustainability itself as an architectural quality attribute (e.g., consider software energy awareness [13], for example), there exists no definition currently available which matches software sustainability and the context of AI Software and its operations, which is the major objective of this paper.

B. AI Software and MLOps At a Glance

According to the definitions of NESSI[4]—the European Technology Platform (ETP) dedicated to Software, Services and Data—AI Software is intended as the integration "of AI components in the smart software systems required to advance the digital transformation", with the identification of four key classes of challenges, namely: "(1) governance of self-adapting software; (2) explainable AI; (3) appropriate data; and (4) legal, ethical and societal challenges" along with a specific set of technical challenges namely: "to (1) re-engineer software technologies, and address (2) software composability; (3) software and data lifecycles; (4) quality assurance; (5) socio- technical challenges; and (6) dedicated hardware". Figure 2 shows an outline of a typical DevOps process (top) mapped to the sub-processes being enacted during DataOps, which feature, one the one hand, an instantiation of the Cross-Industry Standard Process for Data Mining (CRISP-DM) [7] and, on the other hand, the typical tools used as gateways in every-day DevOps processes.

Along the lines defined above, and from a practitioner perspective, there are several emerging software components and ecosystems already available which can be combined into a complex cloud AI software application. For example, as aforementioned, tools such as Apache AirFlow and KubeFlow support lifecycle operations, including data-lifecycle management (e.g., see Conflluent[5]) featuring data Extraction Transportation and Load as well as explainability (e.g., see the Google Explainability Center [6]).

However, the aforementioned software components constitute stove-piped solutions, which hardly integrate between themselves, if at all. Also, the algorithm design principles entailed in such software components are often themselves supported by individual software components and data-intensive middleware. For example, consider Figure 3 which shows a Lifelong Hybrid Learning algorithm tailored from Pal et al. [14].

The figure shows the algorithm as being oriented towards processing in a knowledge-base (KB), enacting an endless loop, operating semantically following windowed semantics, operating featuring a queueing system such as Apache Kafka and so on. Each of these design principles are supported by their own data-intensive middleware (e.g., Apache Kafka for queuing and reliable message-passing).

Along the same vein, and from a research perspective, AI software research is concentrating in discovering or improving modelling techniques (e.g., towards deep learning operations [15]) as well as improving software operations across emergent compute situations (e.g., Edge computing

[4]http://www.nessi-europe.com/default.aspx?page=home
[5]https://www.confluent.io
[6]https://cloud.google.com/explainable-ai

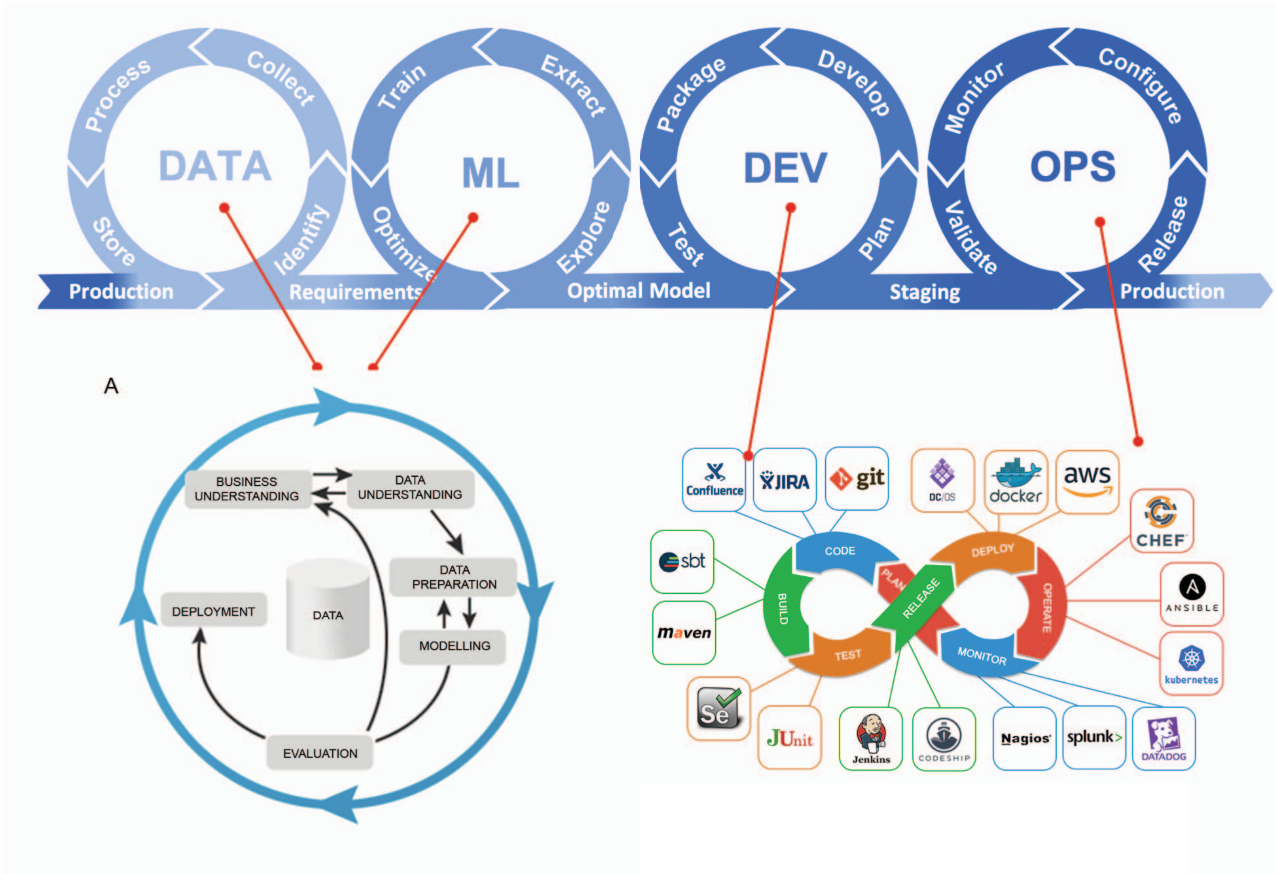

Figure 2. MLOps, a recap featuring a typical DevOps process model.

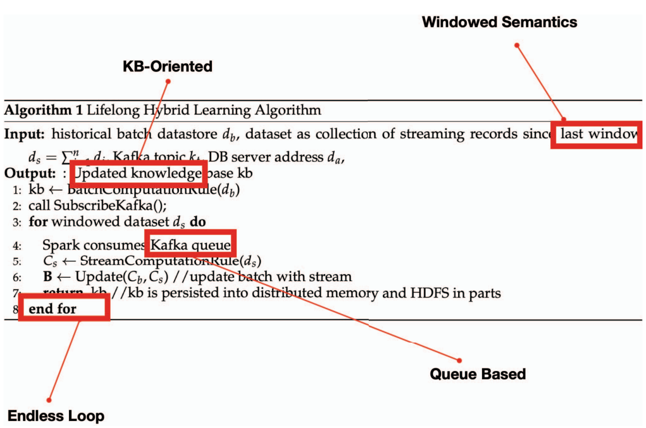

Figure 3. AI Software Algorithms, a sample from Pal et al.[14]; design principles are highlighted.

and consequent Federated Machine-Learning [16]). Similarly, MLOps research is in its infancy and provides definitions [17] and primordial research into the support for actual operations as well as AI software quality assurance [18].

III. Challenges in AI Software Operations: An Educational Perspective

This section offers an overview onto the organisational and educational structures around AI Software operations and the gaps thereof. Such an overview is essential to fully grasp the needs and ends of the conceptualisation later offered for sustainable MLOps. The section starts from the limitations perceived at an educational perspective and offers an overview over how, at the enterprise level, MLOps and AI software organisations are limited in supporting such operations. Challenges from an educational perspective reside in offering a steady flow of AI software professionals capable of incepting, maintaining, and continuously evolve AI software during operations as part of a business value chain [19].

1) Context and Data: The insights recapped in this section draw from data and experiences reported at the Jheronimus Academy of Data Sciences (JADS) in The Netherlands during its 4-year lifespan as a key institute of excellence in training AI Entrepreneurs, a considerable portion of which—around 80% in fact—end up becoming responsible for specific AI operations and data pipelines in industry. JADS is the unique and recent (since 2016) collaboration of two renowned research universities—namely, the Eindhoven University of Technology and Tilburg University—bringing together top academics to further the interdisciplinary academic field of data science and AI engineering as well as entrepreneurship in any of the two or both.

JADS as an institution aims to build two bridges: (1) the bridge between (digital) technology and the social sciences; (2) the bridge between academia and society. JADS offers nine academic educational programs (Bachelor, Master, PDEng, PhD, and Professional Education) and three research centers and has seen some 1000+ students in these programs so far.

The data and statistics used in the rest of this section draw from the pool of evaluation and trend analysis over the operations at JADS over the years.

2) Challenges in Educating for AI Operations - The JADS Experience: The data at hand, reflects five facts:

Fact 1. **Over 90% of industries *are* data-intensive but do not know what to do with their data.** JADS reported this experience while discussing with the many data-owners who sponsor theses internships for JADS students. Over a time period of 4 years, 90% of internship providers reported that they had no indoor data strategy for their business and were indeed considering to prepare one.

Fact 2. **75% data "scientists" are *not* computer scientists by training.** On the one hand, the definition of Data Science itself is a complex endeavour still lacking a rigorous investigation. On the other hand, JADS studentship draws from a multitude of backgrounds and indeed, less than 20% of such a studentship yearly reflects young laureates in Computer Science Bachelors Degrees.

Fact 3. **For those companies that have indeed embraced a data strategy, 88% of their data science teams are short of Data Engineers by a factor of 2.** Industries in the JADS ecosystem report that the fully-specialised data-engineering talent in their human-resources is understaffed, with an overwhelming majority of data-scientists surpassing the presence of data engineering professionals skilled in the ways of data-intensive continuous computing and DataOps [2]. As a result of this fact, the same talent reports that—in 66% of the times— they are overworked and are unlikely to continue in their position should the work-pressure remain the same.

The facts above yield a rather grim picture.

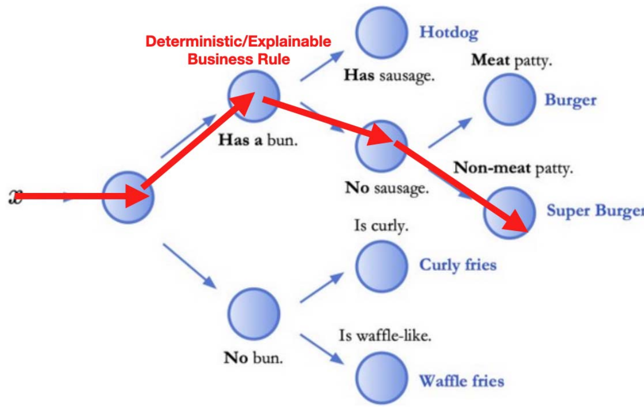

Figure 4. Decision-trees are explainable, *by-design*; an example of a business rule (in red) easily elicited by reading the tree in an if-then-else fashion from top to bottom.

Summary. Not only the educational output is limited, but the expertise-level yielded is even more insufficient and does not match the expectations of industry both in terms of quantity of graduates and their qualities/skills, with a strong emphasis on the engineering angle.

As an anecdotal evidence of this conclusion stand the statements, for example, of the Google Cloud Training directorate who recaps the point in late 2019, paraphrasing: "With the market for artificial intelligence and machine learning-powered solutions projected to grow to $1.2B by 2023, [...] our customers have witnessed internally that the data engineering role has evolved and now requires a larger set of skills [...]".

IV. Trends in AI Software Operations

From an opposite perspective, AI software is typically designed in a domain-driven fashion and focuses on addressing specific business- or mission-intelligence concerns. Concerning the trends around the operations of such domain-driven AI software, they all rotate around guaranteeing a set of specific properties to be supported by such software while operating for the domain it was designed for. At a glance, such properties reflect four frontiers:

1) **Explanation.** Since 25th May 2018, the General Data Protection Regulation (GDPR) [20] establishes a right for all individuals to obtain "meaningful explanations of the logic involved" when "automated (algorithmic) individual decision-making"—including profiling—takes place. Under this condition, AI Software designers and operators need to offer guarantees that they can back-track their own operations from the initial inception of a data-point to the emission of the automated decision. On the one hand,

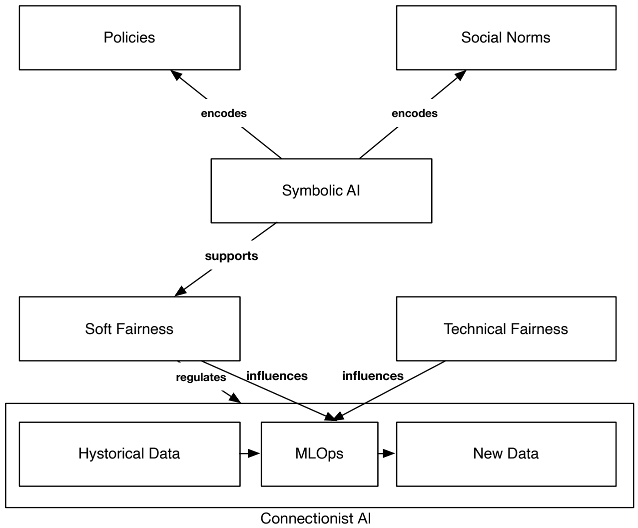

Figure 5. fairness in ML Ops requires a combined interaction between symbolic and connectionist AI, drawn from [24].

there exist Machine-Learning algorithms which are *by-design* explainable such as decision-trees [21], which offer their prediction in the form of easily reverse-engineerable business rules (see Fig. 4, a business-rule is highlighted in red). On the other hand, complex AI software typically includes tens of autonomous decision-making components choreographed as part of an equally complex orchestration of business services (e.g., think of the series of Natural-Language Processes enacted during customer-service operations alone) which cannot be trivially *monitored* [22]—or even further—*observed* [23].

2) **Fairness.** Upon the definition of fairness, rests one of the major contact-points between the two major schools of thought in AI, namely, *symbolic*, and *connectionist* [25]. First, fairness reflects a multitude of software properties—e.g., the extent to which a software gives fair allocation of decision power to a group of arbiters or stakeholders [26]—which typically individually map to a specific domain. Under this condition, an AI software would reflect a continuity between its operations and the intended social contract beneath such operations. Figure 5 offers a non-exhaustive recap of AI operations fairness from a very high level of abstraction. On the one hand, *soft* fairness arises from jurisdictional and social sciences research, addressing and enforcing—through regulatory policies and social norms—equality and equal opportunity for all data points at any quality level. On the other hand, technical fairness reflects the mitigation of bias and discrimination is commonly done by either obfuscating sensitive features in the data (process fairness [27],grappiolo), by introducing additional costs/performance metrics during training (outcome fairness [28]), or by appending further decisional thresholds to a biased model (post-processing fairness [29], [24]).

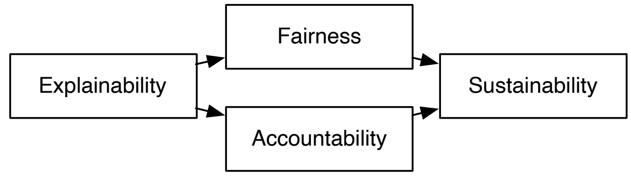

Figure 6. Towards sustainable MLOps, a cascade of AI Software properties to be maintained during operations to warrant for sustainability.

3) **Accountability.** In the day and age of COVID-19, machine-learning software accountability has taken an unexpected turn. Not only does decision-making software need to support explanation of its actions, but that same software is required to withstand liability from its owners, (ab-)users, and original designers for any feature which is misaligned to the social norms under which it was designed to operate. For example, in connection with the recent pandemic, AI software to trace contagions as well as several connected MLOps have undergone severe press coverage because of their short-sightedness towards accountability such as UK's contact-tracing app being used for stalking and without possibility for it and its creators to account for such a failed feature-use[7].
4) **Sustainability.** Lastly, sustainability of ML operations entails the contemporary assurance of all the above frontiers, since: (1) explanation leads MLOps to be observable and self-improvable as well as possibly continuous with their intended social contract; (2) explainability leads to fairness which is the foundation for sustaining said social contract itself; (3) accountability accompanies fairness matching it with the legal establishment in which any MLOps are entailed. Figure 6 offers a recap of such a definition.

V. Conclusions

This paper offers a recap of the trends and challenges around the pursuit of sustainable machine-learning operations, where sustainability is intended as a multi-faceted characteristic of a software system capable of autonomous decision-making or prediction. The paper explores the state of the art from a bird's eye perspective. Subsequently, the paper takes an educational perspective offering a deeper dive into the needs and shortcomings of the AI talent market, exposing the issues it features. Finally, the paper offers a preliminary exploration of the challenges thus outlined and offered a research roadmap in their pursuit. All in all, the

[7]https://nakedsecurity.sophos.com/2020/05/14/woman-stalked-by-sandwich-server-via-her-covid-19-contact-tracing-info/

several scientific communities involved in the subject matter may take inspiration from the contents and material in this paper to pursue their own research agendas in the directions defined above.

Acknowledgment

The author's work is partially supported by the European Commission grant no. 787061 (H2020), ANITA, European Commission grant no. 825040 (H2020), RADON, European Commission grant no. 825480 (H2020), SODALITE.

References

[1] T. M. Mitchell, *Machine learning*. New York, NY [u.a.: McGraw-Hill, 2010. [Online]. Available: http://www.amazon.com/Machine-Learning-Tom-M-Mitchell/dp/0070428077

[2] A. R. Munappy, D. I. Mattos, J. Bosch, H. H. Olsson, and A. Dakkak, "From ad-hoc data analytics to dataops." in *ICSSP*. ACM, 2020, pp. 165–174. [Online]. Available: http://dblp.uni-trier.de/db/conf/ispw/icssp2020.html#MunappyMBOD20

[3] P. Lipton, D. Palma, M. Rutkowski, and D. A. Tamburri, "Tosca solves big problems in the cloud and beyond!" *IEEE Cloud Comput.*, vol. 5, no. 2, pp. 37–47, 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/cloudcomp/cloudcomp5.html#LiptonPRT18

[4] G. Lawton, "Making business intelligence more useful," *Computer*, vol. 39, pp. 14– 16, 2006. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=35942&arnumber=1703302

[5] P. Heck and G. Schouten, "Turning software engineers into ai engineers." *CoRR*, vol. abs/2011.01590, 2020. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr2011.html#abs-2011-01590

[6] G. Casale, M. Artac, W.-J. Heuvel, A. Hoorn, P. Jakovits, F. Leymann, M. Long, V. Papanikolaou, D. Presenza, A. Russo, S. Srirama, D. Tamburri, M. Wurster, and L. Zhu, "Radon: rational decomposition and orchestration for serverless computing," *SICS Software-Intensive Cyber-Physical Systems*, vol. 35, pp. 77–87, 2019.

[7] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0 step-by-step data mining guide," The CRISP-DM consortium, Tech. Rep., August 2000. [Online]. Available: http://www.crisp-dm.org/CRISPWP-0800.pdf

[8] M. Razavian, G. Procaccianti, and D. A. Tamburri, "Four-dimensional sustainable e-services." in *EnviroInfo*, J. M. Gmez, M. Sonnenschein, U. Vogel, A. Winter, B. Rapp, and N. Giesen, Eds. BIS-Verlag, 2014, pp. 221–228. [Online]. Available: http://dblp.uni-trier.de/db/conf/enviroinfo/enviroinfo2014.html#RazavianPT14

[9] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," *IEEE Software*, vol. 33, no. 3, pp. 94–100, May 2016.

[10] J.-J. Rousseau, *The Social Contract*, ser. Penguin Great Ideas. London: Penguin, 2004.

[11] R. C. Seacord, J. Elm, W. Goethert, G. A. Lewis, D. Plakosh, J. E. Robert, L. Wrage, and M. Lindvall, "Measuring software sustainability." in *ICSM*. IEEE Computer Society, 2003, pp. 450–. [Online]. Available: http://dblp.uni-trier.de/db/conf/icsm/icsm2003.html#SeacordEGLPRWL03

[12] C. Calero, M. F. Bertoa, and M. . Moraga, "Sustainability and quality: Icing on the cake." in *RE4SuSy@RE*, ser. CEUR Workshop Proceedings, B. Penzenstadler, M. Mahaux, and C. Salinesi, Eds., vol. 995. CEUR-WS.org, 2013. [Online]. Available: http://dblp.uni-trier.de/db/conf/re/re4susy2013.html#CaleroBM13

[13] R. Kazman, S. Haziyev, A. Yakuba, and D. A. Tamburri, "Managing energy consumption as an architectural quality attribute." *IEEE Softw.*, vol. 35, no. 5, pp. 102–107, 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/software/software35.html#KazmanHYT18

[14] G. Pal, G. Li, and K. Atkinson, "Multi-agent big-data lambda architecture model for e-commerce analytics." *Data*, vol. 3, no. 4, p. 58, 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/data/data3.html#PalLA18

[15] B. Ulker, S. Stuijk, H. Corporaal, and R. Wijnhoven, "Reviewing inference performance of state-of-the-art deep learning frameworks." in *SCOPES*, S. Stuijk and H. Corporaal, Eds. ACM, 2020, pp. 48–53. [Online]. Available: http://dblp.uni-trier.de/db/conf/scopes/scopes2020.html#UlkerSCW20

[16] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications." *ACM TIST*, vol. 10, no. 2, pp. 12:1–12:19, 2019. [Online]. Available: http://dblp.uni-trier.de/db/journals/tist/tist10.html#YangLCT19

[17] W.-J. van den Heuvel and D. A. Tamburri, "Model-driven ml-ops for intelligent enterprise applications: Vision, approaches and challenges," in *Business Modeling and Software Design*, B. Shishkov, Ed. Cham: Springer International Publishing, 2020, pp. 169–181.

[18] C. Tao, J. Gao, and T. Wang, "Testing and quality validation for ai software?perspectives, issues, and practices," *IEEE Access*, vol. 7, pp. 120 164–120 175, 2019.

[19] D. K. Yoo and J. J. Roh, "Value chain creation in business analytics." in *HICSS*, T. Bui, Ed. ScholarSpace / AIS Electronic Library (AISeL), 2018, pp. 1–10. [Online]. Available: http://dblp.uni-trier.de/db/conf/hicss/hicss2018.html#YooR18

[20] D. A. Tamburri, "Design principles for the general data protection regulation (gdpr): A formal concept analysis and its evaluation." *Inf. Syst.*, vol. 91, p. 101469, 2020. [Online]. Available: http://dblp.uni-trier.de/db/journals/is/is91.html#Tamburri20

[21] X. Hu, C. Rudin, and M. I. Seltzer, "Optimal sparse decision trees." in *NeurIPS*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alch Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 7265–7273. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2019.html#HuRS19

[22] D. A. Tamburri, M. Miglierina, and E. D. Nitto, "Cloud applications monitoring: An industrial study." *Inf. Softw. Technol.*, vol. 127, p. 106376, 2020. [Online]. Available: http://dblp.uni-trier.de/db/journals/infsof/infsof127.html#TamburriMN20

[23] D. A. Tamburri, M. M. Bersani, R. Mirandola, and G. Pea, "Devops service observability by-design: Experimenting with model-view-controller." in *ESOCC*, ser. Lecture Notes in Computer Science, K. Kritikos, P. Plebani, and F. D. Paoli, Eds., vol. 11116. Springer, 2018, pp. 49–64. [Online]. Available: http://dblp.uni-trier.de/db/conf/esocc/esocc2018.html#TamburriBMP18

[24] A. Meyer-Vitali, R. Bakker, M. van Bekkum, M. de Boer, G. Burghouts, van Diggelen, D. J., G. J., d. G. C., H. J., and S. A., Raaijmakers, "Hybrid ai: A whitepaper," *TNO Reports*, 2019. [Online]. Available: https://repository.tno.nl//islandora/object/uuid:510e0747-30f4-4af4-9447-9330b7bd4b47

[25] P. Smolensky, "Connectionist ai, symbolic ai, and the brain." *Artif. Intell. Rev.*, vol. 1, no. 2, pp. 95–109, 1987. [Online]. Available: http://dblp.uni-trier.de/db/journals/air/air1.html#Smolensky87

[26] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building classifiers with independency constraints." in *ICDM Workshops*, Y. Saygin, J. X. Yu, H. Kargupta, W. Wang, S. Ranka, P. S. Yu, and X. Wu, Eds. IEEE Computer Society, 2009, pp. 13–18. [Online]. Available: http://dblp.uni-trier.de/db/conf/icdm/icdmw2009.html#CaldersKP09

[27] M. S. Qafari and W. M. P. van der Aalst, "Fairness-aware process mining." in *OTM Conferences*, ser. Lecture Notes in Computer Science, H. Panetto, C. Debruyne, M. Hepp, D. Lewis, C. A. Ardagna, and R. Meersman, Eds., vol. 11877. Springer, 2019, pp. 182–192. [Online]. Available: http://dblp.uni-trier.de/db/conf/otm/otm2019.html#QafariA19

[28] B. Fang, M. Jiang, P. yi Cheng, J. Shen, and Y. Fang, "Achieving outcome fairness in machine learning models for social decision problems." in *IJCAI*, C. Bessiere, Ed. ijcai.org, 2020, pp. 444–450, scheduled for July 2020, Yokohama, Japan, postponed due to the Corona pandemic. [Online]. Available: http://dblp.uni-trier.de/db/conf/ijcai/ijcai2020.html#FangJCSF20

[29] P. K. Lohia, K. N. Ramamurthy, M. Bhide, D. Saha, K. R. Varshney, and R. Puri, "Bias mitigation post-processing for individual and group fairness." in *ICASSP*. IEEE, 2019, pp. 2847–2851. [Online]. Available: http://dblp.uni-trier.de/db/conf/icassp/icassp2019.html#LohiaRBSVP19