NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696 Website: ijiert.org
VOLUME 7, ISSUE 11, Nov.-2020

AN EXPLORATORY STUDY ON WHY AI LIFECYCLE MODELS NEED TO BE REVISED

Sudhir Allam,
Sr. Data Scientist, Department of Information Technology, USA

ABSTRACT
This paper examines why AI life cycles need to be revised to suit recent advances in technology and consumer needs. The findings from the research show that the current AI lifecycle models are facing many challenges that can be addressed by revising their states and approaches of how they are operationalized or deployed. The paper outlines the fact that AI lifecycle stages have been ignored by existing lifecycle models. This involves the collection of data, feasibility analysis, reporting, model surveillance, and model risk evaluation [1]. This research also shows that the real complexities of implementing artificial intelligence go far beyond advanced learning algorithms - more attention needs to be given to the entire life cycle. In particular, irrespective of current artificial intelligence development tools, it is noted that they still do not meet specific features of this area. As they progress toward AI, the majority of companies develop data science teams comprised of individuals knowledgeable about AI algorithms, processes, and techniques [1]. However, many of those organizations, instead of getting the initiatives into full operation and incorporated with current applications and processes, are struggling to make their AI projects genuinely applicable to companies. This is why so many stakeholders in the industry see only a small proportion of AI projects as genuine success. Clients from across industries recognize quickly that they need a systemic "operationalization" solution to AI to drive AI performance [2]. This method ensures that the entire AI end-to-end life cycle is revised and managed.

Keywords: AI lifecycle, Software development, machine learning, software engineering

INTRODUCTION
Artificial Intelligence (AI) is becoming more and more essential for businesses. The next artificial intelligence breakthrough is being embraced by tech-led organizations. Smart systems replace and cooperate with conventional components of the software. The need for model reproductivity, traceability, and verification of the company leads to improvements in the conventional AI lifecycle [2]. Recent access to massive data volumes and increased computational capacity have exploded the range of scenarios in which AI can be implemented. In reality, AI is now used in critical organizational environments to add value. Consequently, there are a series of unforeseen challenges in the life cycle of AI models, from development to deployment and application [2]. These projects should be designed, reviewed, debugged, implemented, managed, and incorporated into complex systems, just as standard software applications [2]. For highly regulated industries, this is especially significant because new processes must be developed to ensure that AI systems satisfy all the requirements [3]. The differences between developing AI systems and developing standard Software Engineering systems have been the subject of a recent study. Owing to the rapid rate of development in AI and recent advances in Software Engineering, these lifecycle models have flaws when implemented in the industry. To address this, the study aims to recognize and improve how the existing industry faces the challenges of designing machine learning applications [3]. The research findings would reveal important problems which should be tackled when applying artificial intelligence lifecycle models. Feasibility evaluations, reporting, risk model assessment, and model management are phases that have been ignored by current lifecycle models [4]. The requirements are lacking, and the documentation and management of machine learning models must be automated. Eventually, this paper lays the groundwork for the revision of AI lifecycle models to meet the industry's current needs.

RESEARCH PROBLEM

The main problem this exploratory study seeks to resolve is why AI Lifecycle models need to be revised to build AI systems other than regular software engineering systems. Given the rapid rate of AI changes and recent advances in software engineering, there are shortcomings in these lifecycle models. Companies at the forefront of AI development are trying to reinvent their development methods and developing innovative technologies. There are also things to be learned to support other organizations and guide research towards the industry.

LITERATURE REVIEW

A. Significant gaps in the current AI lifecycles models

Although data scientists' access to technology and resources has increased significantly, the data science lifecycle has remained stagnant. The earliest releases of CRISP-DM produced 20 years ago and the latest life cycles from leading suppliers such as Google, Microsoft, and DataRobot have not changed much [4,5]. The majority of the data science lifecycle versions also cover the same range of tasks: complex business awareness, domain data understanding, acquisition and engineering information, model creation and skills development, model development, and monitoring [6]. However, business needs have changed as data science is integrated into most firms. Model replicability, regulatory compliance, and verification have now become a basic necessity for data science in large businesses [7]. Unsurprisingly, these conditions are often ignored or understated in leading AI lifecycle models.

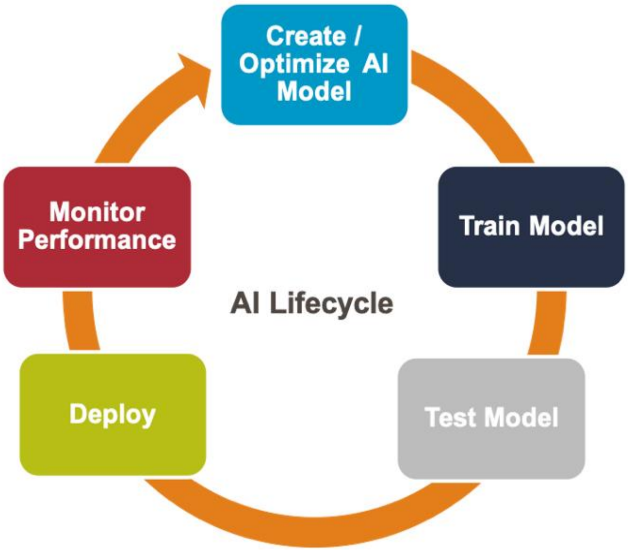

Fig i: AI lifecycle

B. Why should model life cycles of AI be revised?

An overwhelming majority of data scientists have concentrated too much on the inner cogs of algorithms, forgetting that the main thing is that they have the right information in the right position at the right time. Unless one is working for an AI venture or powerhouses like Uber, Facebook, Google, etc, the algorithms currently deployed on Keras, PyTorch, Theano, TensorFlow, MXnet are likely that can shape the problem [8]. The construction of a scalable and stable data pipeline that ends with a General Linear Model or a Random Forest allows several businesses to acquire value [9]. Companies with hundreds of data scientists create various AI models to do incredible work, ending with wonderful once-in-a-lifetime models, which take 3-12 months to look at the output [9]. Moreover, the fact that artificial intelligence (usually) focuses upon the tuning and data pipelines of hyperparameters does not indicate that the cycle needs to be restored or a whole new method is being looked for. A strong foundation lies in the market and the AI momentum: cultural shift to promote experimental research, ongoing assessment, networking, integration, reliability, and the work of goods and services [9].

C. How Artificial Intelligence Affects Software Development

To comprehend why AI lifecycle models should be revised, one must first comprehend the role of AI in software development by examining what can be modified. These are some of the main features provided by AI in the creation of software to give the end-users severe custom products or services.

1. Software design

In the preparation and design of software projects to provide a specific approach, it takes professional learning and experience. Setting the right configuration for each step must be a designer-prone process [10]. Dynamic configuration modifications are made for the recalls and potential investigation plans before the customer arrives at the desired solution. By automating some complex processes utilizing artificial intelligence software, users can enable the most efficient approaches to secure designs [10]. A good example is the AI Design Assistant (AIDA) in which many developers need to understand the customer's needs and expectations before developing a bigger design. AIDA operates as a web development method that enables examine various software design combinations by providing the required customized solution to customer needs.

2. Automatically generate code

Taking an n organization’s ideas and coming up with a code for the big project can be time-consuming. Time and money will be saved by developers by coming up with a solution that helps write code before beginning production. With the complexity of what the aim is to gather certain information that can be very time-consuming as you compose the code from scratch. AI-based support eliminates these loads to a degree by automating code creation and detecting code errors [11]. A good example is a project where the concept can be understood in the natural language and framework by translating it into executable code [11].

3. AI-oriented research

Software testing is a critical phase of software production since it helps guarantee the product's consistency. If a certain program test is replicated regularly, the source code should be modified by running the same checks, which is more time-intensive and costly. Several software development tools incorporate AI to build test cases and do regression testing [12]. Most of these AI methods contribute to the automation of research services, ensuring error-free testing. Artificial intelligence and machine learning-focused research tools such as Testim.io, Functionalize, and Appvance is only a few examples [12].

The AI lifecycle presents the same issues as standard production does, albeit from a different angle: version management, packaging, deployment, coordination, and serving [13]. The primary concern is that we are attempting to impose strategies that were previously utilized in software creation on this ecosystem. There has been a significant increase in the number of technologies (especially Open Source) attempting to fix the development lifecycle of AI—Spark takes full advantage of Kubernetes, kubeflow, MLflow, and cloud providers providing resources to develop and serve models [13]. We're experiencing this entire ecosystem growth and development and increasing climate [14].

D. The AI-bugs

In terms of AI bugs, it is important to be aware of three types of dataset bugs which are bias, drift, and fragility. Bias arises from bias in the datasets used to create the function and can have disastrous consequences, particularly if used on BlackBox-like models (Hello, DeepLearning!) — Amazon abolishing the 'sexist AI' model was the result of an AI lifecycle model that met all checks in an organization renowned for its quality engineering efficiency [14]. When attempting to sort and hire computer programmers, nevertheless, data is skewed. That meant the algorithm disadvantaged female applicants as there were not many cases on the training set. This will also occur in mortgage rating services and several other companies. Weapons of Math Destruction is a 2016 book that posed many of these issues in algorithms making critical decisions about recruiting, classifying individuals, and others—it's a great read!

Drift arises when models are designed, well-functioning, and deployed [14]. To maintain consistency, the model should be recalibrated and resynchronized per application and data. Otherwise, it would continue to deteriorate with time. Fragility may be attributed to bias but more to developments beyond the control of the team. The worst thing is that the bulk of these AI bugs are difficult to identify before output — for Bias, there are many forms, but not many since it is seen as a sluggishness element. Therefore, control and observability, other Deployment foundations, play a gigantic role in AI lifecycles. Users must evaluate proxies identifying the business benefit it should affect the AI components [15]. The AI components can not explicitly be tracked; however, proxy measurements can be analyzed. Such forms of metrics will help identify and fix the AI bugs quickly.

E. The distance between IT and data scientists

One of the main challenges that Data science teams encounter each day, namely the gap between the enterprise and the real industrialization/operationalization of what has been built, is the same issue that has plagued the corporate community and the Automation revolution. This gap becomes clearer and more obvious as companies start to shift their attitude to software creation and distribution. Why does this happen? Since several corporate successes are apparent in introducing innovative practices, tools and achieving the toughest thing: transforming the community [15]. It is not easy to increase sharing and cooperation; it is what every company should achieve most: transform the community. Any cultural influences can have a great effect in the case of AI engineers and data scientists, but most convincing [15]. Most of them have a highly scholarly experience, but they spend a long time focusing on one issue until it is sufficiently good for a publication to approve. The bar is exceedingly large not only about those parameters but also concerning the nature of the tests, the statistical rigor, and so forth [16]. This is significant in a market sense, but not so... This ensures that it is okay to publish and have the concept in a deployable state, with 40 percent precision. It is cheaper to have that ready than to wait months to get anything "good enough" to get it in development. Maybe this would no longer be a worth solving challenge in three months [16]. The easiest way to move quickly, with customizable option

In the initiation of an AI initiative, the first essential move is to scope and to choose appropriate application instances to be dealt with in the AI model. During this process, it is important that the strategic business goals and desired project results can be clearly defined, all priorities of the stakeholders are aligned, main tools and measures anticipated and performance measurements defined. The choice of AI or machine learning cases and the ability to test ROI is crucial to the progress of a data project.

The next stage in the machines lifecycles is the Design or Build process, which, based on the type of the task, will last from several days to some months, until the projects concerned are properly chosen and targeted. The design stage is an iterative method that involves all the steps necessary to construct the AI or machine learning model: acquiring, exploring, preparing, purifying, engineering, evaluating, and running a range of models and try to anticipate actions or find information. It is crucial for the progress of the model building that all the persons participating in the AI project have access to data, resources, and processes to collaborate in various stages [16]. Model validation is another important aspect to consider: How can the output of each iteration be determined, measured, and evaluated concerning the ROI target defined?

Artificial intelligence frameworks must not be on the shelves with actual business benefits from data projects; they must be implemented or installed in development and be used in the whole enterprise. Throughout this point, ROI is once again a crucial consideration: it is necessary to remember that it is not possible or appropriate to make all AI projects functional [16]. The costs of installing a concept in development are often greater than the benefit it brings. In the project scoping process, this can ideally be expected before the concept is fully constructed, although this cannot always be achieved. The replicability of a project can be seen as an important consideration in the implementation process of an AI lifecycle: imagine how this project may be replicated and capitalized by other teams, agencies, countries, etc. [16].

FUTURE IN THE U.S

AI would certainly influence the way we construct the application and experience it better in the United States. Artificial intelligence is undoubtedly going to influence the landscape of software growth in the United States, as the companies get increasingly curious regarding AI technologies. The overarching concern, though, is whether artificial intelligence will help to subvert the human need to improve technologies. There is no question about the enormous growth we are seeing for Computer vision solutions that help to simplify components of the artificial intelligence model simple technique, increasing the workflow of data scientists and allowing technology providers to train production-quality models [18]. There are several AI-driven applications available on the market, such as Sagemaker, Google Cloud AutoML, and Driverless AI's H20.ai, which are designed for automation or standardizing of key components of database planning, model searching, and tuning, model rollout, and scaling [18]. Generally speaking, the architecture and development of the program appear to be affected by artificial intelligence. The developers of applications must take advantage of the possible advantages of AI as a software creation game-changer.

ECONOMIC BENEFITS TO THE U.S

Revising AI lifecycle models has economic benefits to U.S companies in terms of how they conduct their operations. More sophisticated technologies are being developed to make the task easier. It is projected that approximately 80% of businesses invest in AI and 47% of digitally well-developed businesses define AI strategies. Artificial intelligence software can generate a market valuation of $2.9 billion for potential forecasts. In the United States, preparation for deployment and a strong fractional rise in replacement AI are the major drivers of their economic impact, representing the region's leading position on AI and its deployment, as well as a high automation opportunity projected to occur at the regional level by 2030. AI is economically influenced by (a) efficiency benefits from automating procedures, and by the expansion by AI technology (aided, independent and enhanced intelligence) of the current labor force; and (b) improved market demand arising from the provision of personalized and/or better-quality AI-enhanced goods and services [19]. A projected 58 percent of the GDP impact in 2030 will be derived from side effects on demand or $9.1 million of extra GDP. Over the entire period 2017-2030, however, the productivity gains would account for approximately 55% of the GDP effect. The faster (total) drive-in process on the output side of the economy is reflected, since the consumption-side effects of the GDP depend heavily on the delayed indirect impact of competitive enterprise, increasing the availability of personalized, high-quality AI items and making them more accessible [19].

CONCLUSION

This paper outlined why AI lifecycles models need to be revised considering the many challenges that it already faces. A key takeaway of AI is that its models use vast amounts of data to refine their internal configuration to categorize it according to existing data as new data are introduced. Humans aren't very good at working with vast amounts of data, and the overwhelming amount of data accessible to us often keeps us from accessing it actively. Although AI proposals are historically called the three key phases of AI scoping, development, and operationalization, this work does not stop with the implementation of the model. For the achievement of true Enterprise AI, models must be monitored while in development, and those new models can be implemented rapidly, tested, trained, and implemented to shift tactics or adjust to shifting conditions daily — the most widely known stage for data science and machinery learning [17,18]. Finally, users need people with various expertise to help each stage in the AI life-cycle to execute an effective AI strategy. Therefore, any aspect of the lifecycle of AI models requires to be dissected, converted into specific organizational capabilities and needs, and then mapped to the various data profiles.

[1]

REFERENCES

1) D. Partridge, "Engineering artificial intelligence software", Artificial Intelligence Review, vol. 1, no. 1, pp. 27-41, 1986.
2) L. Melli, "Experiences of Programming Languages for Artificial Intelligence Research: A Case Study", INFOR: Information Systems and Operational Research, vol. 15, no. 1, pp. 107-129, 1977.
3) R. Plant and S. Murrell, An executive's guide to information technology. Cambridge: Cambridge University Press, 2007.
4) H. Rozenfeld, E. Zancul and V. Rodrigues, "Insights on software selection based on reference models for business processes: an example in product lifecycle management", International Journal of Product Lifecycle Management, vol. 12, no. 2, p. 131, 2019.
5) Sharon and D. Dori, "Model-Based Project-Product Lifecycle Management and Gantt Chart Models: A Comparative Study", Systems Engineering, vol. 20, no. 5, pp. 447-466, 2017.

6) S. Wellsandt, E. Nabati, T. Wuest, K. Hribernik and K. Thoben, "A survey of product lifecycle models: towards complex products and service offers", International Journal of Product Lifecycle Management, vol. 9, no. 4, p. 353, 2016.
7) E. Zancul, V. Rodrigues and H. Rozenfeld, "Insights on software selection based on reference models for business processes: an example in product lifecycle management", International Journal of Product Lifecycle Management, vol. 12, no. 2, p. 131, 2019.
8) Sommerville, "Artificial intelligence — Applications in the Future of Software Engineering", Artificial Intelligence in Engineering, vol. 2, no. 4, p. 235, 1987.
9) O. Etzioni, "Readings in AI and software engineering", Artificial Intelligence in Engineering, vol. 4, no. 2, p. 104, 1989.
10) Möller, "Lifecycle models of data-centric systems and domains", Semantic Web, vol. 4, no. 1, pp. 67-88, 2013.
11) Gomez, J. Bedolla, F. Ricci, and P. Chiabert, "Validation process model for product lifecycle management", International Journal of Product Lifecycle Management, vol. 7, no. 23, p. 230, 2014.
12) Hribernik, K. Thoben, T. Wuest, S. Wellsandt and E. Nabati, "A survey of product lifecycle models: towards complex products and service offers", International Journal of Product Lifecycle Management, vol. 9, no. 4, p. 353, 2016.
13) Brom, O. Belova and A. Sissinio, "Lifecycle Costs for Energy Equipment: FMECA & Lifecycle Costing Models as “Decision Making” Tools for Cost Reduction During the Whole Equipment Life", Procedia Engineering, vol. 152, pp. 173-176, 2016.
14) Ruparelia, "Software development lifecycle models", ACM SIGSOFT Software Engineering Notes, vol. 35, no. 3, pp. 8-13, 2010.
15) J. Gomez, J. Bedolla, F. Ricci, and P. Chiabert, "Validation process model for product lifecycle management", International Journal of Product Lifecycle Management, vol. 7, no. 23, p. 230, 2014.
16) Karkasinas and A. Rentizelas, "Verification and validation: a novel product lifecycle management-based approach", International Journal of Product Lifecycle Management, vol. 12, no. 2, p. 149, 2019.
17) R. Ibrahim and B. Paulson, "Discontinuity in organizations: identifying business environments affecting the efficiency of knowledge flows in Product Lifecycle Management", International Journal of Product Lifecycle Management, vol. 3, no. 1, p. 21, 2008.
18) S. Schulte, "Customer-centric PLM: integrating customers' feedback into product data and lifecycle processes", International Journal of Product Lifecycle Management, vol. 3, no. 4, p. 295, 2008.
19) J. Jupp, "Cross-industry learning: a comparative study of product lifecycle management and building information modeling", International Journal of Product Lifecycle Management, vol. 9, no. 3, p. 258, 2016.