Position Paper: Quality Assurance in Deep Learning Systems

Domingos F. Oliveira[1,2][a] and Miguel A. Brito[2,3][b]

[1]*Department of Informatics and Computing, Mandume Ya Ndemufaio University, Lubango, Angola*

[2]*Algoritmi Centre, University of Minho, Guimarães, Portugal*

[3]*Information Systems Department, University of Minho, Guimarães, Portugal*

Keywords: Deep Learning, Software Quality Assurance, Software Quality Assurance Standards, Quality Assurance in DL Systems.

Abstract: The use of DL as a driving force for new and next-generation technological innovation plays a vital role in the success of organisations. Its penetration in almost all domains requires improving the quality of such systems using quality assurance models. It has been widely explored in DM and SD projects, hence the need to resort to methodology like KDD, SEMMA and the CRISP-DM. In this way, the reuse of standards and methods to guarantee the quality of these systems presents itself as an opportunity. In this way, the position paper has the fundamental objective of giving an idea about the form of a structure that facilitates the application of quality assurance in DL systems. Creating a framework that enables quality assurance of DL systems involves adjusting the development process of traditional methods since the challenge lies in the different programming paradigms and the logical representation of DL software.

1 INTRODUCTION

Organisations frequently change their business requirements due to their needs. Technologies, one of the components of Information Systems (IS), play an essential role in changing these requirements.

The use of Deep Learning (DL), as a technique of Machine Learning (ML), and this one of the areas of Artificial Intelligence (AI), has presented advances in solving organisational problems, contributing to solutions such as intelligent systems, assisted reality, autonomous vehicles, medical diagnostics, and fundamentally in solutions for Data Mining (DM), more specifically in Data Science (DS), among others. Moreover, for (Ma et al., 2018), the penetration of DL in almost all domains revolutionises our daily lives, leading to opportunities for study on how to improve the assurance and quality control of such systems.

Taking into account the problems associated with quality assurance (QA) and quality control (QC), to lead to the success of IS in organisation with forme states (Delone and McLean, 1992), it was defined as a central research question for the present work: **how to guarantee quality in DL system?** To answer the question, that is according to the problem presented, it has been defined as the objectives of this position paper to give and make an analysis of some norms of guaranteeing the quality of systems, as well as to present and analyse some methodologies of systems development that use DL.

To meet this objective, we have defined the methodology for this work as a search for bibliographic references about relevant and current jobs in information systems, software engineering and computer science, which deal with the theme of quality assurance in DL systems.

The present position paper comprises four sections, dismissed as follows. The first section is the *Introduction*, the second section the *Study of the Art*, the third will make a *General Approach* about the quality assurance standards and some DL systems development methodologies, as well as we present our idea about a quality assurance framework in DL systems. Finally, we will finish with the fourth section, corresponding to the general *Conclusions* of this work.

[a] https://orcid.org/0000-0002-2890-0655

[b] https://orcid.org/0000-0003-4235-9700

Oliveira, D. and Brito, M.
Position Paper: Quality Assurance in Deep Learning Systems.
DOI: 10.5220/0011107100003269
In *Proceedings of the 11th International Conference on Data Science, Technology and Applications (DATA 2022)*, pages 203-210
ISBN: 978-989-758-583-8; ISSN: 2184-285X

2 STATE OF ART

The DL technique is beneficial in DM and DS design; hence to make a study on quality assurance in systems of the kind, one must present the concepts and a set of norms and standards for quality assurance of techniques, as well as the development methodologies for DM and DS design, offered by the industry.

2.1 Software Quality Assurance (SQA)

The concept associated with software quality is often difficult to present. Still, the same goes through effective quality management applied to create a valuable product that provides measurable value for producers' users, according to (Pressman and Maxim, 2020) and also referenced by the (Sommerville, 2016). For this fact to be evidence, it is necessary to make the management of it.

According to (Institute, 2017), project quality management is a process that determines quality policies, objectives and responsibilities to meet the needs for which it was performed and, supports continuous improvement activities, goes through a plan. System projects that use DL must also go through the same phases. Otherwise, they will not meet customer needs. That is why (Suryn, 2014) stated that quality had become a critical attribute of software products because its absence produces financial, material and sometimes life losses.

Hence (Sommerville, 2016), consider that software quality is not simple, and the result of good project management pass high standards, which involves software engineering methods, management techniques and software quality control and assurance actions.

Quality control is one of the essential activities for management. Its concept relates to a set of activities designed to assess the quality of a developed product, according to (Galin, 2018). It ensures that the product complies with what was stipulated, emphasizing the result and leaving a gap in the process because it is only possible to have a final quality product if the procedure for its development is rigorous as advocated by (Society, 2014).

To obtain a quality product, you must have a rigorous process. To speak of a quality process is to say of assurance, as a set of activities that define and evaluate the suitability of methods, for our case a, software, to provide evidence that establishes confidence that the processes are appropriate and produce products of adequate quality for their intended purposes, this according to (Society, 2014).

While (Galin, 2018), considers SQA as a set of systematically planned actions required to provide adequate confidence that software development conforms to established functional technical requirements and also to the management requirements of meeting deadlines and operating on a budget.

In this way is that the quality of software, if present as fundamental for systems considered critical or that involve DL and help teams to create these systems that meet the needs of users, this according to a set of standards planned and systematic actions required to support this need, such as (Pressman and Maxim, 2020).

According to (Suryn, 2014), quality has become a critical attribute of products as its absence produces financial, health and sometimes life losses. QA involves a set of activities. Therefore, it should be employed in another set of activities being the software development process or in the entire development cycle.

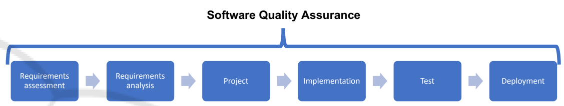

Figure 1: The perspective of QA in the IS development cycle adapted from (Suryn, 2014).

Figure 1 is a simple demonstration of what has been addressed so far about how the quality assurance process should be represented in the development cycle of a system. Thus we consider quality assurance to be necessary, as it establishes an infrastructure that supports sound software engineering methods, project management, and QC actions, all of which are essential to building software, according to (Pressman and Maxim, 2020).

For this, the *Plan-Do-Check-Act (PDCA)* can present itself with a fundamental element for the quality assurance process since the same is a quality management system, which allows the continuous improvement of processes and products through a flow of activities, used to pay attention and adjust the deviations that may occur in the process, as the figure 2 this according to (Isniah et al., 2020).

Software quality has been an approach in system projects for a long time, but when it comes to DL systems, it is still a dilemma, according to (Ma et al., 2018). In this way, we first seek to understand the norms and standards that make QA possible on systems.

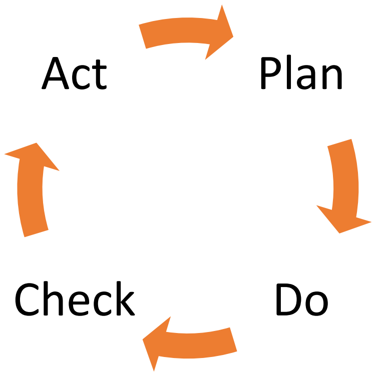

Figure 2: PDCA adapted from (Isniah et al., 2020).

2.2 Software Quality Assurance Standards

Quality standards focus on quality assurance, depending on the system and concentrate first on accepting or responding to user needs. Their use of means leads to some benefits, according to (Galin, 2018) development and maintenance, better understanding and mutual coordination between development and maintenance teams.

Still, (Galin, 2018), in the last two decades, there has been a rapid development of international SQA standards. This is due to the increased coverage of related topics, which has led to a greater understanding of the standards and their need. As a result, four standards have had wide acceptance in the community, namely ISO 9000-3, ISO/IEC 12207, ISO/IEC 15504, and CMMI.

2.2.1 ISO 9000-3

The ISO 9000 - 3 arises from other standards such as ISO 9000, being a series of standards for quality management, which allows for identifying errors and streamlining operation, provides a design, development, production, installation and services used as criteria to qualify development organizations for contracts and gives a significant impact to the software industry (Inoue et al., 1994).

2.2.2 ISO/IEC 15504

ISO/IEC 15504 is a set of standards which propose models to improve and evaluate processes related to IS and software products (Patón-Romero et al., 2018), allowing the improvement of its development process.

2.2.3 CMMI Standard

CMMI provide guidance, facilitating the development of solutions by improving your ability to manage product or service development, acquisition and maintenance. It has three categories, CMMI-SVC for services, CMMI-ACQ for purchase, and CMMI-DEV for growth, which focus on best practices for developing quality products/services that meet or exceed customer expectations (Ayyagari and Atoum, 2019).

2.2.4 ISO/IEC 12207

The standard, ISO/IEC 12207, is a framework that defines a software life cycle process, with well-defined terms that can be referenced in the software industry according to (Anwer et al., 2018).

2.3 Comparison of Quality Assurance Standards

For a better analysis, table 1 was prepared to compare the standards, considering an overview approach, considering five perspectives: scope, focus, compatibility, implementation model, and process.

Table 2 presents the advantages and disadvantages of quality assurance standards.

An approach has been made to the norms and standards of quality assurance, and the assumption of seeking to apply quality assurance to DL systems should be taken into account. To this end, the following sections describe the concepts of DL systems. In addition, I am trying to understand its scope and way of development to analyse how its quality should be guaranteed.

2.4 Quality Assurance in DL Systems

According to (Santhanam et al., 2019), recent advances in AI using DL techniques have triggered their large-scale use in a wide range of applications, plus the current levels of maturity, of using an AI component in applications considered critical can have unexpected consequences, leading to concerns about their reliability, repeatability, trustworthiness, and maintainability. In this way, we intend to approach the concepts of employing QA in DL systems.

2.4.1 Deep Learning Systems

DL has emerged as a new research area in ML (Deng and Yu, 2013), allowing computational models composed of several processing layers to learn data representations with various levels of abstraction, making

Table 1: Comparison of quality assurance standards adapted from (Konttinen, 2016).

	ISO 90003	ISO/IEC 15504	CMMI	ISO/IEC 12207
Scope	General	Software development process	Software development process	Software development process
Focus	Customers and processes	Process assessment	Business and process improvement	Framework for software life cycle processes
Compatibility	CMMI level 3 compliant	CMMI and ISO 90003 compliant	ISO 15504 compliant	
Implementation	Full and flexible compliance is required	Flexible continuous improvement model	Focused, phased and continuous improvement models	————————
Approach	Verification of documented standards.	Evaluates the project on capability levels.	Evaluates maturity levels.	Focuses on procurement, delivery, operation and maintenance.

Table 2: Advantages and disadvantages of quality assurance standards.

Standards	Advantages	Disadvantages
ISO 9000-3	Is independent of the life cycle, technology, processes and organizational structure.	It is not used as an evaluation criterion in certifications.
ISO/IEC 15504	Provides a framework for assessment, maintains the improvement process, assesses risks, and determines capabilities.	It is a more comprehensive model, has its complexity, does not define a precise assessment method.
CMMI	Maturity levels; Process improvement; On-time delivery; increased customer satisfaction; quality culture in the programmers.	Overload on documentation, time/effort to implement, culture changes and does not integrate with other models.
ISO/IEC 12207	To help organizations understand the components present in the acquisition and delivery of software effectively.	It only provides a framework of software processes, activities, and tasks that can be identified, planned, and executed.

information available in traditional and new scopes and extending key aspects of artificial intelligence (Lecun et al., 2015).

Thus, it is essential to understand its concept to study quality assurance in systems of this nature.

According to (Johnson, 2021), DL systems can be considered as systems that use multi-layered neural networks to perform learning tasks, including regression, classification, clustering, and automatic coding, applying ML techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation and pattern analysis and classification, following a set of 5 phases.

The application of these techniques means that DL systems are highly exploited nowadays. Hence quality assurance is essential, as they can be classified as critical systems and require more excellent care in their use.

This is why (Xiao et al., 2018), consider that the fundamental characteristics of DL systems involve feedback-driven exploration, where a user performs a set of tasks to get the best result for a specific mission based on a set of characteristics.

These features are associated with models that go through Convolution Neural Network (CNN), Deep Belief Networks (DBNs), Recurrent Neural Network (RNNs) and Self Encoders (AE), where each of these is used to solve a specific problem, as they can be trained in a supervised, unsupervised and reinforcement manner.

The characteristic is related to the explanatory variable, and its extraction involves a set of data that lacks greater precision. Hence, the DL systems development process is very delicate, and this leads us to look at the DL systems development life cycle, as presented in figure 3.

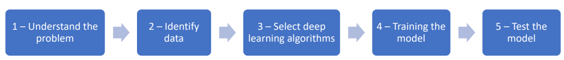

Figure 3: DL process adapted from (Johnson, 2021).

2.4.2 DL Systems Development Process

According to (Zhang et al., 2019a), (Zhang et al., 2019b), DL system development adopts a different programming paradigm and practice from conventional software applications. Hence, software engineering for traditional system development cannot become effective for DL system development.

The figure 4 presents a simple demonstration of the disparity that exists between the conventional application development process and the process used for the development of DL systems, adapted from (Simard et al., 2017) and (Amershi et al., 2019), as, for DL systems, it is presented as some fundamental data layers and lack training.

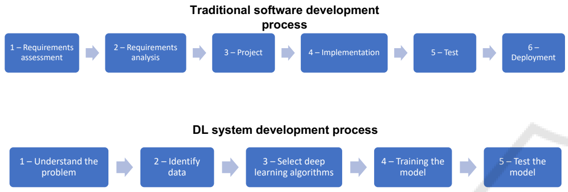

Figure 4: Conventional development process vs DL systems development process.

In this way, it becomes fundamental to make an in-depth analysis of the same development process aimed at the quality assurance of DL systems since it acts on a set of activities that form the same process since DL systems differ from traditional ones. However, the difference is visible in different activities.

Quality assurance has long been addressed for conventional systems. For DL systems, it becomes a challenge due to problems that can cause misuse and avoiding normal accidents that technologies sometimes cause. Proof of this are the various works (Masuda et al., 2018), (Ma et al., 2018), (Nakajima, 2018), (Liu et al., 2019), (Nakajima, 2019), , (Fujii et al., 2020) that place this area as a challenge for the scientific community and professionals.

There are some works that make an approach on DL systems development processes (Amershi et al., 2019), (Zhang et al., 2020), (Zhang et al., 2019a), (Zhang et al., 2019b), where they consider that it, adopts a data-driven programming paradigm and practice. Hence, doing an analysis on DM/DS projects (Foroughi and Luksch, 2018) is fundamental using methodologies.

2.5 Methodologies for Developing DL Systems

The management of DM and DS projects is a complex task; hence a methodology is needed, as the development process of these projects adopts the methods like KDD (Fayyad et al., 1996), SEMMA (Olson and Delen, 2008), and CRISP-DM (Schröer et al., 2021).

2.5.1 Knowledge Discovery in Databases - KDD

It refers to the overall process of knowledge discovery from data, using an extraction process that applies specific algorithms for extracting patterns from data (Fayyad et al., 1996). This methodology comprises steps for knowledge discovery from the existence of data, and DM is one of the steps in this process (Fávero, 2019).

2.5.2 Sample, Explore, Modify, Model, and Access - SEMMA

Developed by the SAS Institute (Institute, 2021), it starts with a statistically representative sample of data, facilitating the application of exploratory statistics and visualisation techniques to select and transform the most significant predictive variables to the model, predicts outcomes and assesses the model accuracy, this as stated (Olson and Delen, 2008).

2.5.3 Cross-Industry Standard Process for Data Mining - CRISP-DM

It is a process model that describes the life cycle of DS projects, allowing planning, organising, and implementation of a project involving ML or DL. It consists of six phases with arrows indicating the most important and frequent dependencies between steps (Wehrstein and Bachmann, 2020).

2.6 Comparison of DL Systems Development Methodologies

We present a comparison in table 3, taking into account some attributes of the KDD, SEMMA and CRISP-DM methodologies, to give a better perspective on their use in projects involving the development of DL systems, more specifically in the area of DM and DS, taking as a starting point the work of (Molero-Castillo et al., 2018).

Table 4 gives an overview of the strengths and limitations and the advantages and disadvantages of KDD, SEMMA and CRISP-DM methodologies.

By the table 4, it is possible to notice that the CRISP-DM presents itself better because it allows its stages can be reversed, facilitating the correction of possible errors without having to finish the whole cycle this as it affirms (Pyvovar et al., 2019). Still, it

Table 3: Comparison of KDD, SEMMA and CRISP-DM methodologies adapted from (Molero-Castillo et al., 2018), (Dåderman and Rosander, 2018).

Description	KDD	SEMMA	CRISP-DM
Purpose	Extract hidden knowledge from data	Guide the implementation of data mining applications	Create a reliable, incremental process for delivering value
Strategy	Map low-level data to other forms that may be more compact, more abstract, or more valuable	Provide an easy to understand the process, allowing for organised project development and maintenance	Make software development lightweight
Phases	——— Data selection Data pre-processing Data transformation Data mining Interpretation and Evaluation ———	——— Sample Exploration Modification Model Evaluation ———	Business knowledge Data understanding Data preparation Modelling Evaluation Implementation

Table 4: Advantages and disadvantages of the KDD, SEMMA and CRISP-DM methodologies.

Methodologies	Advantages	Disadvantages
KDD	It is iterative (Molero-Castillo et al., 2018).	It does not describe the tasks and activities that should be performed in each phase (Molero-Castillo et al., 2018). Leaves gap in interpretation and visualization (Dåderman and Rosander, 2018).
SEMMA	It is an iterative process, focuses on data management and DM model aspects, supports user and various DM techniques (Dåderman and Rosander, 2018)	It does not describe the activities carried out in each phase (Molero-Castillo et al., 2018). Ignores the evaluation phase of the work, designed to work with SAS Enterprise (Dåderman and Rosander, 2018).
CRISP-DM	Division of phases, tasks and activities, iterative, supported on DM techniques and has documentation (Dåderman and Rosander, 2018). Stages can be reversed, as any problem can be corrected without interfering with the cycle (Pyvovar et al., 2019)	Does not include control, monitoring, communication, knowledge management or quality, change and team management activities, ignores maintenance or updating of models (Michalczyk and Scheu, 2020). Lacks some maturity model (Molero-Castillo et al., 2018). Long and complicated process (Dåderman and Rosander, 2018). Difficulty in preparing and modelling streaming data (Kalgotra and Sharda, 2016).

presents with disadvantages the fact of not providing a sound vision of management on communication, knowledge and aspects of quality management as it affirms (Michalczyk and Scheu, 2020). Hence it is fundamental to our idea.

2.7 Focus on Quality Assurance in DL Systems

Assuming that a DL system is the application of techniques on a set of data to be trained to learn by itself through patterns in several layers of processing, their quality assurance involves applying a set of parameters to the development process of these systems to have a product that meets customer needs.

Proof said is what they considered (Ma et al., 2018), where they defined DL systems assurance as to the accuracy that training data plays in modelling the learning process and decision logic of systems and that the same systematic approach lacks quality control, inspection and evaluation as the same data can be introduced as malicious.

In our view, quality assurance should be seen throughout the development process of DL systems to eliminate possible errors that can already be detected in the initial phase of the process itself.

3 APPROACH

As we have presented above, the DM and DS project methodologies are fundamental to the DL system development process. Hence, associated with PDCA may be helpful in the implementation of quality assurance of these systems. In this way, our perspective on quality assurance of DL systems, based on the combination of the same, is our focus.

3.1 Quality Assurance Perspective in DL Systems

Given the various opinions on quality assurance in DL systems, we present our vision on a proposal to create a framework that can assist the process of developing DL systems safely and with excellent quality. Our concept aims first to define a set of activities that allow the development of these systems with quality.

The model devised by us is based on the use of two tools already proven and widely used in industry and academia, which is the use of PDCA, applied to CRISP-DM, giving a framework perspective on quality assurance in DL system as illustrated by 5, which presents itself as an opportunity to be used in DM and DS projects, as is the case of CRISP-ML(Q) (Studer et al., 2020).

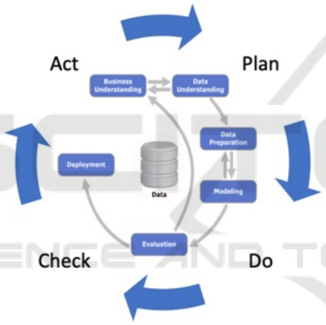

Figure 5: CRISP-DL(Q).

We intend to develop a framework called CRISP-DL(Q), which will be applied to the development of DL systems, enabling quality assurance, combining CRISP-DM and PDCA to overcome the limitations presented by CRISP-DM, which does not include quality management.

4 CONCLUSIONS

The present position paper approaches SQA and SQC, where some standards for software development process are presented, namely ISO 90 003, ISO/IEC 15 504, CMMI and ISO/IEC 12 207, their advantages and disadvantages, as well as a comparison of them. Also, we present the concepts of DL systems, where one of the most significant applications is in the development of DM and DS projects, using methodologies such as KDD, SEMMA and the CRISP-DM, which were also presented and analysed.

We end the document by presenting a perspective on our idea about a framework that enables quality assurance of DL systems, all by the literature. In this way, the development and implementation of a quality assurance framework for DL systems, namely CRISP-DL(Q), presents a challenge for future work to be developed.

ACKNOWLEDGEMENTS

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

REFERENCES

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., and Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, pages 291–300.

Anwer, S., Wen, L., Rout, T., and Wang, Z. (2018). Introducing requirements change management process into ISO/IEC 12207. *Communications in Computer and Information Science*, 918:185–199.

Ayyagari, M. R. and Atoum, I. (2019). CMMI-DEV Implementation Simplified: A Spiral Software Model. *International Journal of Advanced Computer Science and Applications*, 10(4):445–450.

Dåderman, A. and Rosander, S. (2018). Evaluating Frameworks for Implementing Machine Learning in Signal Processing: A Comparative Study of CRISP-DM, SEMMA and KDD. Technical report.

Delone, W. H. and McLean, E. R. (1992). The quest for the dependent variable. Information Systems Research. *Information System Research*, 3(1):60–95.

Deng, L. and Yu, D. (2013). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3-4):197–387.

Fávero, L. P. (2019). KDD e Data Mining: mais do que apenas conceitos.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *American Association for Artificial Intelligence*, 17(3):37–54.

Foroughi, F. and Luksch, P. (2018). Data Science Methodology for Cybersecurity Projects. *5th International Conference on Artificial Intelligence and Applications*, pages 1–14.

Fujii, G., Hamada, K., Ishikawa, F., Masuda, S., Matsuya, M., Myojin, T., Nishi, Y., Ogawa, H., Toku, T., Tokumoto, S., Tsuchiya, K., and Ujita, Y. (2020). Guidelines for Quality Assurance of Machine Learning-Based Artificial Intelligence. *International Journal of Software Engineering and Knowledge Engineering*, 30(11-12):1589–1606.

Galin, D. (2018). *Software Quality: Concepts and Practice*. John Wiley & Sons, Inc., 1th edition.

Inoue, K., Watanabe, A., Iida, H., and Torii, K. (1994). Modeling Method for Management Process and Its Application to CMM and ISO9000-3. *Proceedings of the Third International Conference on the Software Process. Applying the Software Process*, pages 1–14.

Institute, P. M. (2017). *A Guide to the Project Management Body of Knowledge*. Project Management Institute, Pennsylvania, 6th edition.

Institute, T. S. (2021). About SAS.

Isniah, S., Hardi Purba, H., and Debora, F. (2020). Plan do check action (PDCA) method: literature review and research issues. *Jurnal Sistem dan Manajemen Industri*, 4(1):72–81.

Johnson, D. (2021). Deep Learning Tutorial: Neural Network Basics for Beginners.

Kalgotra, P. and Sharda, R. (2016). Progression analysis of signals: Extending CRISP-DM to stream analytics. *2016 IEEE International Conference on Big Data*, pages 2880–2885.

Konttinen, V. (2016). Towards Disciplined Software Development: ISO 9001:2008 Based Software Process Improvement in SME. Technical Report May, Helsinki Metropolia University of Applied Sciences.

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Liu, Y., Ma, L., and Zhao, J. (2019). Secure Deep Learning Engineering: A Road Towards Quality Assurance of Intelligent Systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11852 LNCS:3–15.

Ma, L., Juefei-Xu, F., Xue, M., Hu, Q., Chen, S., Li, B., Liu, Y., Zhao, J., Yin, J., and See, S. (2018). Secure Deep Learning Engineering: A Software Quality Assurance Perspective. *arXiv*, (Dl):1–11.

Masuda, S., Ono, K., Yasue, T., and Hosokawa, N. (2018). A survey of software quality for machine learning applications. *11th International Conference on Software Testing, Verification and Validation Workshops*, pages 279–284.

Michalczyk, S. and Scheu, S. (2020). Designing an Analytical Information Systems Engineering Method. *Twenty-Eighth European Conference on Information Systems (ECIS2020)*, (June).

Molero-Castillo, G., Bárcenas, E., Sánchez, G., and Antonio-Aquino, A. (2018). User-centered data mining tool for survival-mortality classification of breast cancer in mexican-origin women. *Intelligent Data Sensing and Processing for Health and Well-being Applications*, (January):223–242.

Nakajima, S. (2018). [Invited] Quality Assurance of Machine Learning Software. *7th Global Conference on Consumer Electronics (GCCE 2018)*.

Nakajima, S. (2019). Quality Evaluation Assurance Levels for Deep Neural Networks Software. *2019 International Conference on Technologies and Applications of Artificial Intelligence*.

Olson, D. L. and Delen, D. (2008). *Advanced Data Mining Techniques*. Springer-Verlag Berlin Heidelberg.

Patón-Romero, J. D., Baldassarre, M. T., Rodríguez, M., and Piattini, M. (2018). Green IT Governance and Management based on ISO/IEC 15504. *Computer Standards and Interfaces*, 60(May):26–36.

Pressman, R. S. and Maxim, B. R. (2020). *Software Engineering - A Proctitioner's Approach*. McGraw-Hill Education, New York, 9th edition.

Pyvovar, N., Yamkovyi, K., and Vechirko, M. (2019). Data Science project management methodologies.

Santhanam, P., Farchi, E., and Pankratius, V. (2019). Engineering Reliable Deep Learning Systems. *AAAI Fall Symposium Series on AI in Government & Public Sector*, pages 1–8.

Schröer, C., Kruse, F., and Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, 181(2019):526–534.

Simard, P. Y., Amershi, S., Chickering, D. M., Pelton, A. E., Ghorashi, S., Meek, C., Ramos, G., Suh, J., Verwey, J., Wang, M., and Wernsing, J. (2017). Machine Teaching: A New Paradigm for Building Machine Learning Systems. *arXiv*, pages 1–14.

Society, I. C. (2014). Guide to the Software Engineering Body of Knowledge. *IEEE Computer Society*, pages 1–335.

Sommerville, I. (2016). *Software Engineering*. Pearson Education Limited, 10th edition.

Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., and Mueller, K.-R. (2020). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *arXiv*.

Suryn, W. (2014). *Software Quality Engineering: A Practitioner's Approach*. John Wiley & Sons, Inc, New Jersey.

Wehrstein, L. and Bachmann, B. (2020). CRISP-DM ready for Machine Learning Projects.

Xiao, W., Bhardwaj, R., Ramjee, R., Sivathanu, M., Kwatra, N., Han, Z., Patel, P., Peng, X., Zhao, H., Zhang, Q., Yang, F., and Zhou, L. (2018). Gandiva: Introspective cluster scheduling for deep learning. *3th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2018*, pages 595–610.

Zhang, J. M., Harman, M., Ma, L., and Liu, Y. (2020). Machine Learning Testing: Survey, Landscapes and Horizons. *IEEE Transactions on Software Engineering*, X(X):1–37.

Zhang, T., Gao, C., Ma, L., Lyu, M., and Kim, M. (2019a). An Empirical Study of Common Challenges in Developing Deep Learning Applications. *30th International Symposium on Software Reliability Engineering, ISSRE*, 2019-Octob:104–115.

Zhang, X., Yang, Y., Feng, Y., and Chen, Z. (2019b). Software Engineering Practice in the Development of Deep Learning Applications. *arXiv*.