Vision Towards Quantum Humanities: An Architecture of a Holistic QML Environment

Johanna Barzen
University of Stuttgart
Institute of Architecture of Application Systems
Stuttgart, Germany
johanna.barzen@iaas.uni-stuttgart.de

***Abstract*—The benefits of using classical computers in the humanities has been proven by the establishment of the digital humanities. Recently, quantum computers reached a state in which advantages can be realized in certain use cases. E.g., initial feasibility studies in the humanities have shown that some advantages can already be achieved, and much more can be expected, especially in applying quantum machine learning. But building quantum applications requires a deep understanding of mathematics, quantum algorithms and programming that cannot be expected by humanists. Consequently, they cannot explore potentially promising applications or apply concrete potentials of quantum algorithms. The proposed architecture and method aim at exactly this, enabling (digital) humanists to easily build their own quantum applications in the focus area of quantum machine learning to allow a broad use and critical evaluation of the potentials of quantum algorithms for humanities research.**

Keywords*—*quantum computing, quantum humanities, quantum machine learning

I. Introduction and Motivation

The establishment of digital humanities proves the benefits of using computers as well as techniques and methods from computer science for research in the humanities [4, 13]. Especially processing big data fast is stated as one of the greatest advantages for the humanities [17]. Also, the amount of data digitally available is constantly increasing [16], thus, a comprehensive analysis of this data and its critical interpretation is essential to the digital humanities [13]. This means – regardless of the discussion of how quantitative results can be embedded in qualitative understanding and interpretation – that the demands and requirements for analysis techniques in terms of their accuracy and usability are constantly growing. Various platforms for humanists provide tools (mostly specialised in the analysis of text, e.g., [28, 36, 50], but also suitable for other topics such as [37]) to explore analysis features in the field of digital humanities.

Quantum computers are rapidly evolving in terms of number of qubit, decoherence times, and better error rates [42]. This allows to solve problems in more and more application areas by quantum applications [21, 34]. Thus, the vision of quantum humanities [7] goes beyond the usage of classical computers and makes use of quantum computers for research in the humanities. But what can be expected by the use of quantum computers – with all the given difficulties that this emerging technology still has [24, 41] – for humanities research? When assuming an "ideal" quantum computer, there are different potentials that need to be taken into account [2]: (i) an (exponential) speed-up of certain algorithms [47], (ii) the processing of large amounts of data (quantum parallelism) [45], (iii) the precision of results for certain problems [20, 48], (iv) the solution of previously practically unsolvable problems [15, 35], (v) the solution of problems that are unsolvable in principle classically [46], (vi) low-cost access relative to the computing power enabled [12], and (vii) energy efficiency [35].

Nowadays, some of these potentials are still theoretical. However, initial feasibility studies have shown that some of them already apply to applications in the humanities. E.g., we showed that in a given set of costume data, the costumes have been classified with higher precision by a support vector machine (SVM) based on quantum kernels than with classical kernels [2, 8]. In the domain of Natural Language Processing (NLP), first attempts have been made to exploit some potentials of quantum computing in the field of humanities research by enabling the development of NLP models explicitly designed for execution on quantum hardware (so-called Quantum Natural Language Processing (QNLP) [27, 31]). Also, the approach of Quantum Computer Music aims at using quantum computers in creating, performing, listening to and distributing music [30].

Quantum computers recently became general available and are easily accessible via the cloud [25]). Today's machines are expected to offer advantages particularly in the area of machine learning [14, 49], which is a technology beneficial to the humanities. Therefore, as one of the core tasks of the digital humanities is reviewing new developments from the IT domain and questioning their "usefulness" for humanities research [22], we want to answer the question: Is it possible and worthwhile to exploit with benefits this emerging technology to address problems from the humanities?

To allow a broad and critical evaluation of these potentials, humanists (i.e., both, humanists and digital humanists) need to be enabled to build their own quantum applications. But currently building such applications is a time-consuming and cost-intensive effort asking for deep mathematical and physical understanding of quantum algorithms as well as programming skills that cannot be expected from researchers in the humanities. As a result, potentially promising applications and new ideas cannot be explored or concrete potentials are not applied. Therefore, the proposed architecture and method has the goal to enable humanists with a focus on machine learning to solve problems by building their own applications making use of quantum machine learning without having to understand the

979-8-3503-2759-5/23/$31.00 ©2023 IEEE
DOI 10.1109/CSCE60160.2023.00270

underlying quantum technology and without having deep programming skills.

II. Related Work and Main Challenges

In the following, four main challenges are identified that need to be addresses.

A. *Low-Code as Enabler for Quantum Humanities: Supporting Creativity in Programming*

Because programming quantum computers and using quantum algorithms is very different from their classical counterparts [51] their use must become as simple as possible. This demands appropriate methods and tools to enable humanists translating their real-world problems into the quantum world. A graphical metaphor to build corresponding applications is considered to be most effective [10, 23] to get rid of the needs to understand all the technical details that programming quantum-based applications requires today. So-called low-code tools (e.g., [29, 38]) are an upcoming technology in the business domain that support non-IT-affine users to solve business problems from their domain with software build by themselves without any support from IT specialists. Challenge 1: A graphical language and corresponding low-code tool hiding idiosyncrasies of quantum computers and quantum algorithms is needed to easily allow non-IT savvy users to exploit quantum computing to solve problems in the humanities

B. *Enabling Reuse: Finding Suitable Solutions*

Many quantum algorithms exist that solve dedicated problems. For a specific problem at hand, several algorithms are available to solve it. It requires a deep understanding to determine which algorithm to use in a given problem context. In order to find proven solutions for a problem in a certain context without assuming such a deep understanding, pattern languages are in use since decades [1, 9]. They are a suitable vehicle to support non-specialists in discovering solutions that are independent of any specific technology. Also, an easy-to-use search capability is needed that allows to simply specify some details about the technology or properties, and that returns the available implementations to be selected. Such a search targets libraries of implementations, for example [40]. Patterns are more suited for discovery, which is more exploratory in nature, while search is more geared towards fast satisfaction of information demand. Both, discovery and search, are the basis for supporting reuse: reuse of knowledge and reuse of implementations. This will provide the key underpinning of enabling humanists in efficiently making use of quantum computing. Challenge 2: Two ways of enabling reuse are needed: reuse of knowledge by means of discovery of quantum algorithms, and reuse of algorithm implementations by means of search capabilities. Furthermore, an overall method should include guidelines about how and when to make use of them.

C. *Ensuring Reproducibility: Building Application Packages Automatically*

Enabling evaluations of the benefits of quantum computers for the humanities is one of the core objectives. Thus, all results must be reproducible and verifiable by the scientific community. This implies that the application used to achieve new insights, but also all collateral IT artefacts like data that contributed to such insights must be made available to the public. This can be done by packaging all the artefacts into a single file and adding information about how to reconstruct the overall environment from the artefacts to be able to execute the application. This additional information is called the topology of the overall system. The collection of artefacts together with the topology can be wrapped into an archive that is shared [4, 54]. Challenge 3: The automatic construction of application archives including their topology is a mandatory prerequisite for reproducing and verifying research results in the digital humanities and quantum humanities.

D. *Maintaining Simplicity: Avoiding Non-Application-Specific Knowledge*

Reuse, i.e., creating applications from pre-existing building blocks, is a well-established method to significantly increase efficiency and productivity in building software. Simplicity of reuse as required by non-IT users is a significant challenge especially in the domain of quantum computing and machine learning. The reason for this is that algorithms in these domains typically need a lot of parameters as input and the semantics of the parameters are often not at all related to the application problem itself (hyperparameters) [11, 18, 39]. This is a major obstruction for reuse in these domains by non-IT users. Challenge 4: Optimal hyperparameters must be determined automatically [FH19] to enable non-IT savvy users to get best results from their applications without having to understand the technicalities behind reused artefacts. This requires corresponding extensions to include hyperparameters of quantum algorithms.

III. Objectives of the Architecture

The overall goal is to enable humanists to benefit from the emerging new technology of quantum computing. This includes developing their own quantum applications that will allow a comprehensive and critical evaluation of the technology and its usefulness for answering existing and new questions in the humanities. The focus is on quantum machine learning because many applications from the digital humanities demand more advanced analysis of their growing amount of data as quantitative basis for qualitative interpretations. Also, quantum machine learning is successfully applicable in certain cases already today. But there is much more to be explored in terms of application areas, and in terms of the benefits of more mature quantum computers very soon. As quantum computers are expected to have ground-breaking impact, knowledge about proper use of this technology has to be build up already today, especially in areas where the gain of its use is very high but the knowledge gap is considered particularly high too.

Several intermediate goals must be achieved in order to successfully exploit the new horizons of quantum humanities: (i) Based on challenge 1, the proposed architecture needs to aim at providing an easy-to-use graphical language and a corresponding tool enabling humanists to build applications that especially exploit quantum computers and implementations of quantum algorithms in the field of quantum machine learning. For this, only very little programming skills and no understanding of quantum technology is required. (ii) Challenge 2 states the need for supporting to find appropriate quantum algorithms and their implementations. (iii) Also addressing

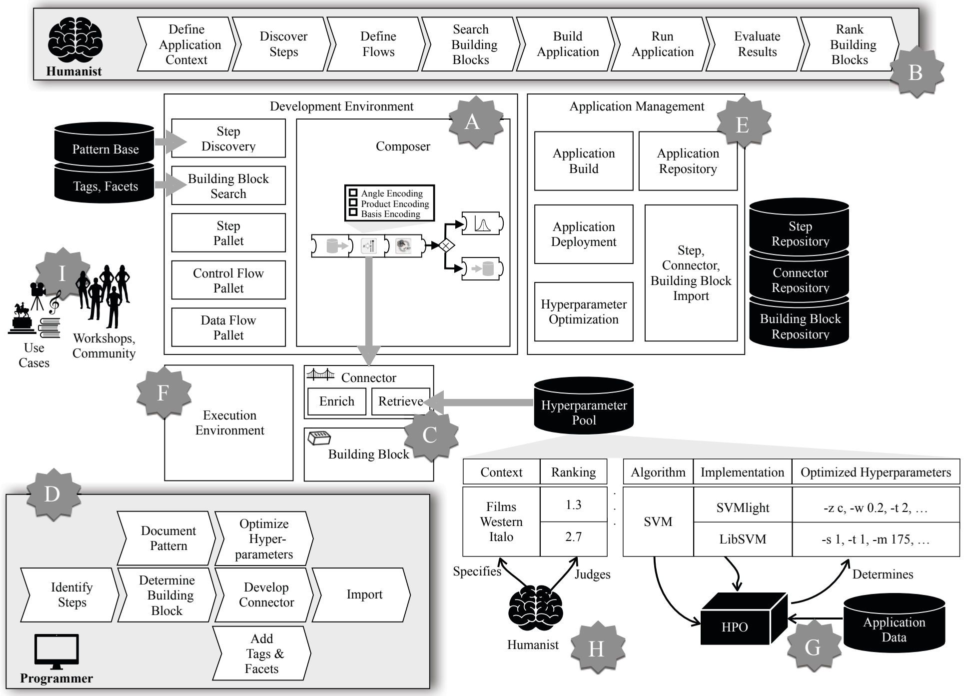

Fig. 1. Overall architecture and method

challenge 2, the goal to be reflected is to provide a method that recommends how humanists can build their own quantum applications based on the proposed environment tackling the issue that complex tools often lack user-friendly instructions of their use, implying that humanists use a trial-and-error approach. (iv) Taking challenge 3 into account, the architecture needs to include automatic construction of application packages together with their topology as prerequisite for reproducing and verifying research results in the digital and quantum humanities. (v) Additionally, addressing challenge 4, a usage mode of quantum machine learning algorithms that frees humanists from understanding detailed parameters expected by the algorithms needs to be reflected. (vi) Both, the architecture and method need to be evaluated by use cases to ensure the suitability for the target users.

IV. Architecture and Method

The overall architecture and method to achieve the stated goals is depicted in Fig. 1. Central is the development environment (A) that allows a graphical composition of quantum applications. For this purpose, steps as graphical abstractions of individual tasks to be performed by the application are dragged to the canvas of the composer and are graphically plugged together. Which steps to be used to solve a particular application problem are suggested to the humanist by a step discovery tool that is part of the development environment. This discovery tool is using knowledge that is stored in a pattern base [26, 52] provided by the tool. A humanist who already knows which step to use can fetch it from the step pallet. Graphically plugging steps means to control the flow in which they are to be performed, i.e. if they are to be performed sequentially, in parallel, conditionally etc.. A control flow pallet provides easy access to such instructions. A step requires input data and produces output data, and the input of one step may be determined from the output of former steps. A data flow pallet offers graphical elements to easily specify such data assignments. While a step describes a task of the application in the abstract, several so-called building blocks are concrete implementations of the task to be performed. If several implementations of a given step are available, a proper one can easily be found via the building block search based on tags or facets that resemble the context of the application.

As Fig. 1 shows, a method (B) is proposed that a humanist can follow to successfully develop a quantum application. The actions of this method need to be extensively described for its users, and its appropriateness has to be verified by use cases with humanists (I). Each application has a context that is defined by a set of tags sketching, for example, the domain of the application, the research questions addressed etc.. By consulting the pattern base, individual steps of the application can be discovered. The control and data flow between the steps are (graphically) defined. Based on the context of the application and preferences of the humanist, a search needs to identify building blocks that implement the steps of the application. Once the application is composed, it has to be build, i.e., all software required is drawn into a self-contained package. Based on this, the application can be run. The humanist can then evaluate the results produced by the application and may decide to change the application. Finally, the building blocks contributing to the result of the application should be ranked: this allows to capture the experience of the humanist made with the building blocks and, thus, help other humanists to search more appropriate building blocks in their application context.

A building block as a concrete implementation of an algorithm may demand parameters that are not perceived as canonical data by a using humanist. Thus, a step may miss to provide data expected by the building block implementing the step. A connector (C) bridges the perception and the demand by automatically enriching the step's data such that they fit a building block's requirements. Special kind of input parameters required by the building block that steer the execution of its implementation (so-called hyperparameters) can be automatically retrieved from a hyperparameter pool. The retrieved parameters need to be optimized for the humanist's current application. The hyperparameter pool stores for each algorithm and for each of its implementations hyperparameters that are optimized for a certain application context (G, H). The application context is specified by a humanist as one action of the method (B). Another action of the method suggests that a humanist should rank the results produced by the building blocks of the application. This application context as well as this rank is associated in the hyperparameter pool with the optimized hyperparameters (H).

Hyperparameters are typically optimized by programmers based on application data of a use case (G). For this purpose, programmers use several of the well-known and already available hyperparameter optimization (HPO) frameworks [19, 53]. An initial set of optimized hyperparameters together with their associated contexts and rankings may result from cases such as [6, 32] and [3, 33]. To allow reproducibility all proposed tools should be made available as open-source with documented interfaces. Thus, optimized hyperparameters can be added to the hyperparameter pool for new algorithms and their implementations at any time by any third party, and contexts and ranking may be added similarly by humanists. Hyperparameter optimization is one of the actions of the method suggested for programmers (D). This method also foresees an action by a programmer (likely jointly with the humanist) to identify new steps required to build an application for a use case. Since steps have a certain graphical appearance that should enable to easily recognize the functionality and use of a step, there is a development effort implied by this action. Steps are quite generic (e.g., store data, prepare a quantum state, cluster a data set etc.), i.e. it is expected that most steps needed to build quantum applications by humanists with focus on quantum machine learning that need be made available. However, new steps can be added to the tools at any time by third parties to extend the tools – even beyond the quantum machine learning area. The method for programmers (D) also contains an action to document best practices about which step to use in order to solve which application problem (i.e., a pattern). This makes up the pattern base used by the step discovery tool of the development environment. Furthermore, implementations of these steps can be determined as building blocks. Application domain specific properties of building blocks can be added by programmers by using tags and facets to enable an easy search of building blocks by humanists when building their own applications. The development of a connector that bridges between a step and its implementing building block (C) is another action to be performed by a programmer. The final action of the method is to import the created artefacts into the application management tool.

The application management tools (E) supports the import of steps, connectors, and building blocks. It stores these artefacts in corresponding repositories. Similarly, complete applications are stored such that they can be retrieved by third parties for verification of the research results produced by the application, or for reuse by other humanists. A complete application is represented by a package that contains all software artefacts required to execute this application. Such an application package is automatically built, i.e. humanists have no need to even be aware that a corresponding package is build. Once an application package has been built, it can be automatically deployed and it is ready for execution. The latter is supported by an execution environment (F) that consists of an engine that interprets the graphical representation of the application logic, that hosts connectors, and that can interact with building blocks in their environment.

V. Current Status and Outlook

For now, the proposed architecture of a comprehensible holistic quantum machine learning environment is still a vision and not all planned components have been implemented yet. Nevertheless, initial steps have been taken to address the challenges outlined and to provide first components for the overall architecture. For example, our tool QHAna, Quantum Humanities Analysis tool [43], provides the basis to the core components envisioned in the overall architecture. QHAna is a plugin-based tool that enable (digital) humanists to build their own machine learning pipelines also making use of quantum machine learning, allowing a comparison of the result of classical and quantum-based machine learning for different application areas in the humanities. Also, the pattern base relies heavily on patterns from the field of quantum computing. This is something my team and I have worked on extensively and already provide several patterns in our Pattern Atlas [44] that offer easy access to quantum computing knowledge. In future work, we plan to continuously add further components to approach the overall vision.

Acknowledgment

This work was partially funded by the BMWK projects PlanQK (01MK20005N).

References

[1] C. Alexander, S. Ishikawa, M. Silverstein. "A Pattern Language: Towns, Buildings, Construction". Oxford University Press 1977.

[2] J. Barzen. "From Digital Humanities to Quantum Humanities: Potentials and Applications". In: E. R. Miranda (ed.). Quantum Computing in the Arts and Humanities: An Introduction to Core Concepts, Theory and Applications, Springer 2022, pp. 1-52. DOI: 10.1007/978-3-030-95538-0_1.

[3] J. Barzen, U. Breitenbücher, L. Eusterbrock, M. Falkenthal, F. Hentschel, F. Leymann. "The vision for MUSE4Music. Applying the MUSE method in musicology". In: H. Engesser (ed.). Computer Science - Research and Development. Advancements of Service Computing: Proceedings of SummerSoC 2016, Vol. 32 (3-4), Heidelberg: Springer, 2016. DOI: 10.1007/s00450-016-0336-1.

[4] S. Bechhofer, D. De Roure, M. Gamble, C. Goble, I. Buchan. "Research objects: Towards exchange and reuse of digital knowledge". In: Nature Precedings, 2010. DOI: 10.1038/npre.2010.4626.1.

[5] D. M. Berry. "Understanding Digital Humanities". Palgrave Macmillan 2012.

[6] J. Barzen, M. Falkenthal, F. Leymann. "Wenn Kostüme sprechen könnten: MUSE - Ein musterbasierter Ansatz an die vestimentäre

Kommunikation im Film". In: P. Bockwinkel et al. (eds.). Digital Humanities. Perspektiven der Praxis. Frank & Timme 2018, pp. 223-241.

[7] J. Barzen, F. Leymann. "Quantum humanities: a vision for quantum computing in digital humanities". In: SICS Software-Intensive Cyber-Physical Systems, 1-6, 2019. DOI: 10.1007/s00450-019-00419-4.

[8] J. Barzen, F. Leymann. "Quantencomputing in den Digital Humanities: innovativ oder übertrieben? Experimente mit quanten-basiertem Maschinellen Lernen". In: Fabrikation von Erkenntnis: Experimente in den Digital Humanities, Zeitschrift für digitale Geisteswissenschaften (ZfdG), 2021. DOI: 10.26298/melusina.8f8w-y749-qidd.

[9] J. O. Coplien. "Software Patterns". SIGS Books & Multimedia 1996.

[10] P. T. Cox. "Visual Programming Languages". In: Wiley Encyclopedia of Computer Science and Engineering, 2008, pp. 1-10. DOI: 10.1002/ 9780470050118.ecse450.

[11] S. Dyrmishi, R. Elshawi, S. Sakr. "A decision support framework for autoML Systems: A meta-learning approach". In: IEEE International Conference on Data Mining Workshops, ICDMW, 2019, pp. 97-106. DOI: 10.1109/ICDMW.2019.00025.

[12] C. Dickel. "A Cloud Quantum Computer Business Plan". In: QuTech, 2018. http://blog.qutech.nl/index.php/2018/07/18/a-cloud-quantum-computer-business-plan/.

[13] J. Drucker. "The Digital Humanities Coursebook. An Introduction to Digital Methods for Research and Scholarship". Routledge 2021.

[14] V. Dunjko, J. M. Taylor, H. J. Briegel. "Quantum-enhanced machine learning". 2016. arXiv: 1610.08251.

[15] D. J. Egger, J. Mareček, S. Woerner. "Warm-starting quantum optimization". 2020. arXiv: 2009.10095v3.

[16] J. Flanders, F. Jannidis (eds.). "The Shape of Data in Digital Humanities Modeling. Texts and Text-based Resources". Routledge 2019.

[17] J. C. Heyder. "Challenging the Copia. Ways to a Successful Big Data Analysis of Eighteenth- Century Magazines and Treatises on Art Connoisseurship". In: S. Schwandt (ed.). Digital Methods in the Humanities. Challenges, Ideas, Perspectives. University Press 2021. pp. 161-180. DOI: 10.14361/9783839454190-006.

[18] F. Hutter, H. Hoos, K. Leyton-Brown. "An Efficient Approach for Assessing Hyperparameter Importance". In: Proceedings of the 31st International Conference on Machine Learning. Vol. 32. Proceedings of Machine Learning Research 1. PMLR, 2014, pp. 754-762.

[19] Overview of hyperparameter optimization tools. https://analyticsindiamag.com/top-hyperparameter-optimisation-tools-neural-networks/.

[20] C. Havenstein, D. Thomas, S. Chandrasekaran. "Comparisons of Performance between Quantum and Classical Machine Learning". In: SMU Data Science Review: Vol. 1: No. 4, Article 11, 2018.

[21] M. Johansson, E. Krishnasamy, N. Meyer, C. Piechurski. "Quantum Computing – A European Perspective". In: Prace: Partnership for Advanced Computing in EuropePartnership for Advanced Computing in Europe, 2021. https://prace-ri.eu/wp-content/uploads/TR-Quantum-Computing-A-European-Perspective.pdf.

[22] F. Jannidis, H. Kohle, M. Rehbein (eds.). "Digital Humanities: Eine Einführung". J. B. Metzler 2017.

[23] M. A. Kuhail, S. Farooq, R. Hammad, M. Bahja. "Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review". In: IEEE Access 9, 2021, pp. 14181–14202. DOI: 10.1109/access.2021.3051043.

[24] F. Leymann, J. Barzen. "The bitter truth about gate-based quantum algorithms in the NISQ era". In: Quantum Sci. Technol. 5 044007, 2020. DOI: 10.1088/2058-9565/abae7d.

[25] F. Leymann, J. Barzen, M. Falkenthal, D. Vietz, B. Weder, K. Wild. "Quantum in the Cloud: Application Potentials and Research Opportunities". In: Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER 2020). SciTePress, 2020, pp. 9-24. DOI: 10.5220/0009819800090024.

[26] F. Leymann. "Towards a Pattern Language for Quantum Algorithms". In: First International Workshop, QTOP 2019, 2019, pp. 218-230.

[27] R. Lorenz, A. Pearson, K. Meichanetzidis, D. Kartsaklis, B. Coecke. "QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer". 2021. arXiv: 2102.12846.

[28] Mallet. https://mimno.github.io/Mallet/index.

[29] Mendix. https://www.mendix.com/de .

[30] E. R. Miranda. "Quantum Computer: Hello, Music!". In: E. R. Miranda (ed.). Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity, Springer 2021. arXiv: 2006.13849.

[31] K. Meichanetzidis, A. Toumi, G. de Felice, B. Coecke. "Grammar-Aware Question-Answering on Quantum Computers". 2020. arXiv: 2012.03756.

[32] MUSE. https://www.iaas.uni-stuttgart.de/en/projects/muse/.

[33] MUSE4Music. https://www.iaas.uni-stuttgart.de/en/projects/muse4music/.

[34] National Academies of Sciences, Engineering, and Medicine: Quantum Computing: Progress and Prospects, The National Academies Press. 2019. DOI 10.17226/25196.

[35] M. Nielsen, I. L. Chuang. "Quantum Computation and Quantum Information". Cambridge University Press 2010.

[36] Natural Language Toolkit. https://www.nltk.org.

[37] Orange. https://orangedatamining.com.

[38] Outsystems. https://www.outsystems.com.

[39] F. Pedregosa. "Hyperparameter optimization with approximate gradient". In: 33rd International Conference on Machine Learning, ICML 2016. Vol. 2. PMLR. 2016, pp. 1150-1159. arXiv: 1602.02355.

[40] M. Piattini, G. Peterssen, R. Pérez-Castillo, J.L. Hevia, M.A. Serrano, et al. "The Talavera Manifesto for Quantum Software Engineering and Programming." In: International Workshop on the QuANtum SoftWare Engineering & pRogramming (QANSWER). 2020, pp. 1-5.

[41] J. Preskill. "Quantum Computing in the NISQ era and beyond". In: Quantum, 2, 79, 2018. DOI: 10.22331/q-2018-08-06-79.

[42] Quantum Computing Report. "IBM Demonstrates 99.9% CNOT Gate Fidelity on a New Superconducting Test Device". 2021. https://quantumcomputingreport.com/ibm-demonstrates-99-9-cnot-gate-fidelity-on-a-new-superconducting-test-device/.

[43] QHAna. https://github.com/UST-QuAntiL/qhana.

[44] Pattern Atlas: Quantum Computing Patterns. https://patterns.platform.planqk.de/pattern-languages/af7780d5-1f97-4536-8da7-4194b093ab1d

[45] E. Rieffel, W. Polak. "Quantum Computing: A Gentle Introduction". The MIT Press 2011.

[46] R. Raz, A. Tal. "Oracle Separation of BQP and PH". In: STOC 2019: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, 2019, pp. 13-23. DOI: /10.1145/3313276.3316315.

[47] T. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, M. Troyer. "Defining and detecting quantum speedup". In: Science Vol. 345(6195), 2014.

[48] D. Sierra-Sosa, J. Arcila-Moreno, B. Garcia-Zapirain, C Castillo-Olea, A. Elmaghraby. "Dementia Prediction Applying Variational Quantum Classifier". 2020. arXiv:2007.08653.

[49] M. Schuld, F. Petruccione. "Supervised Learning with Quantum Computers". Springer 2018. DOI: 10.1007/978-3-319-96424-9.

[50] VOYANT. https://voyant-tools.org.

[51] B. Weder, J. Barzen, F. Leymann, M. Salm, D. Vietz. "The Quantum Software Lifecycle". In: Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software (APEQS 2020). ACM, 2020, pp. 2-9. DOI: 10.1145/3412451.3428497.

[52] M. Weigold, J. Barzen, F. Leymann, M. Salm. "Expanding Data Encoding Patterns For Quantum Algorithms". In: 18th International Conference on Software Architecture Companion (ICSA-C). IEEE, 2021, pp. 95-101. DOI: 10.1109/ICSAC52384.2021.00025.

[53] T. Yu, H. Zhu. "Hyper-Parameter optimization: A review of algorithms and applications". 2020. arXiv:2003.05689.

[54] M. Zimmermann, U. Breitenbücher, J. Guth, S. Hermann, F. Leymann, K. Saatkamp. "Towards Deployable Research Object Archives Based on TOSCA". In: Papers From the 12th Advanced Summer School on Service-Oriented Computing, IBM Research Divisio, 2018, pp. 31-42.

All links were last followed on May 04, 2023.