JYX JYVÄSKYLÄN YLIOPISTO UNIVERSITY OF JYVÄSKYLÄ

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Liimatta, Petri; Taipale, Pauli; Halunen, Kimmo; Heinosaari, Teiko; Mikkonen, Tommi; Stirbu, Vlad

Title: Research Versus Practice in Quantum Software Engineering : Experiences From Credit Scoring Use Case

Year: 2024

Version: Published version

Copyright: © Authors 2024

Rights: CC BY 4.0

Rights url: https://creativecommons.org/licenses/by/4.0/

Please cite the original version:

Liimatta, P., Taipale, P., Halunen, K., Heinosaari, T., Mikkonen, T., & Stirbu, V. (2024). Research Versus Practice in Quantum Software Engineering : Experiences From Credit Scoring Use Case. IEEE Software, 41(6), 9-16. https://doi.org/10.1109/ms.2024.3427168

Editor: **Cesare Pautasso**
University of Lugano
c.pautasso@ieee.org

Editor: **Olaf Zimmermann**
Eastern Switzerland University of Applied Sciences (OST), Rapperswil
olaf.zimmermann@ost.ch

Research Versus Practice in Quantum Software Engineering: Experiences From Credit Scoring Use Case

Petri Liimatta, Pauli Taipale, Kimmo Halunen, Teiko Heinosaari, Tommi Mikkonen, and Vlad Stirbu

From the Editors
The split between classical and quantum computing is becoming more and more relevant for software engineers in computing-intense application domains. Our guest authors share early experiences with quantum-based credit risk scoring in the financial services domain, including a superposition of positive and less positive findings. They highlight key design decisions such as problem partitioning, quantum algorithm selection, and quantum processor selection.

SOFTWARE ENGINEERING RESEARCHERS have picked quantum software as their next target. Techniques that have been applied in classical software engineering on a large scale, such as architectural support or software engineering in general, have been applied in a quantum context, with a series of articles on related topics being published. Moreover, many established software engineering conferences, such as the International Conference on Software Engineering, International Conference on Software Architecture, Foundations of Software Engineering, and Product-Focused Software Process Improvement, are collocated with quantum software workshops, with a separate IEEE event focusing on quantum software (IEEE QSW). The avalanche of venues to report research, as well as the increasing number of articles reporting quantum software engineering research in various forums, seems to indicate that quantum software is becoming a popular research topic, with more and more software engineering researchers contributing to its evolution.

To complement the above research-oriented view, this article approaches quantum software engineering from the viewpoint of industry and its state of practice. The goal is to discuss the level of maturity of quantum software engineering in light of industry expectations from a financial sector use case. As a result, it seems that like

Digital Object Identifier 10.1109/MS.2024.3427168
Date of current version: 10 October 2024

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

in many fields earlier, the increased research has pushed the theoretical limits further but the practical applications seem to be lagging behind. Therefore, understanding the limits and possibilities from the industry point of view is very important.

The Promise of Quantum Computing

Quantum computing is an emerging computation paradigm that relies on the use of qubits (quantum bits), the basic unit of information in these computers. The power of quantum computers relies on the concepts of superposition and entanglement of quantum mechanics that may allow us to solve several problems more efficiently than with classical computers. For example, in weather forecasting, classical nonlinear continuum systems need to be calculated, which is hard for classical computers.[1] A full-scale quantum computer could, at least in theory, solve some computationally demanding problems by using a polynomial number of steps where a normal computer needs an exponentially growing number of steps. The first applications with industrial relevance will most likely be related to optimization problems.

We still are at early phases of the quantum revolution in computations. The present-day gate-based quantum computers consist of fewer than 1,000 qubits, whereas specific purpose quantum annealing[2] computers contain more than 5,000 qubits. Overall, the current generation of quantum computers are known as *noisy intermediate-scale quantum* (*NISQ*) because of the challenges that these devices have.[3,4] NISQ devices are larger than smallest-scale quantum processors with few qubits but are not yet at the scale where error correction methods, like quantum error correction, can be effectively applied. Indeed, the primary challenge on the hardware side is excessively high error rates in results, which requires complex operations simply to decide which result is reliable and which is not.

On the software side, major hindrances are low level of abstraction and the lack of tools for quantum software development.[5] While programming languages have been introduced for quantum computations, composing concrete programs means that the developer is thinking in terms of quantum circuits, which are less abstract than actual quantum algorithms, to a degree that circuits are comparable to circuits in classical computing. Then, the deployment of quantum programs involves several steps where the circuit is first transpiled into a form that can be executed by the particular quantum computer in question, then input to the quantum processor using, e.g., laser pulses as the mechanisms, and finally the result is read and processed to assess its validity. To accomplish the above, classical software is needed to coordinate and refine the quantum executions and to interpret the results, with various interfacing mechanisms that typically are computer-specific.

On the methodological side, the nature of quantum computations is effectively blocking some of the agile principles. In particular, since quantum computations are run in batch mode—where each computation is run from the beginning to the end without any interaction with the user—it is difficult to apply agile methods, even if simulators and other tools can be used to add agility to development and conceptualization.[5] However, since computations may require a long time to complete, as well as consume considerable amounts of energy, there is a limit with respect to how agile the development can be.

From the software development lifecycle perspective,[6] there are certain places in the development process where special considerations for the quantum advantage must be made (Figure 1). To realize the advantage over classical computing alternatives, the following three specific concerns must be met. First, there must be a way to partition the problem, such that a computationally intensive part can benefit from quantum technology. Second, once the partition has been identified, a quantum algorithm is needed that supports the necessary computations. Finally, a quantum processor is needed that is available at the requested time and capable of executing the algorithm in a fashion that provides additional value in the computation. (See the "Quantum Computing Glossary" for a list of terms related to quantum computing.)

Putting Quantum to Practice: Experiences From the Financial Sector

Financial institutions operate within a nexus of complex optimization and stochastic modeling problems, such as portfolio optimization, pricing financial assets, and credit scoring (see "Some Complex Optimization and Stochastic Modeling Problems Such as Portfolio Optimization, Derivative Pricing, and Credit Scoring"). Conventional computational methodologies often struggle to handle these problems accurately in a timely manner, especially when dealing with vast financial datasets and dynamic market conditions.

Quantum computing promises an unprecedented paradigm shift for such financial computations. Already, some quantum algorithms operating on NISQ devices,[7] like quantum amplitude estimation (QAE) and variational quantum eigensolver (VQE),

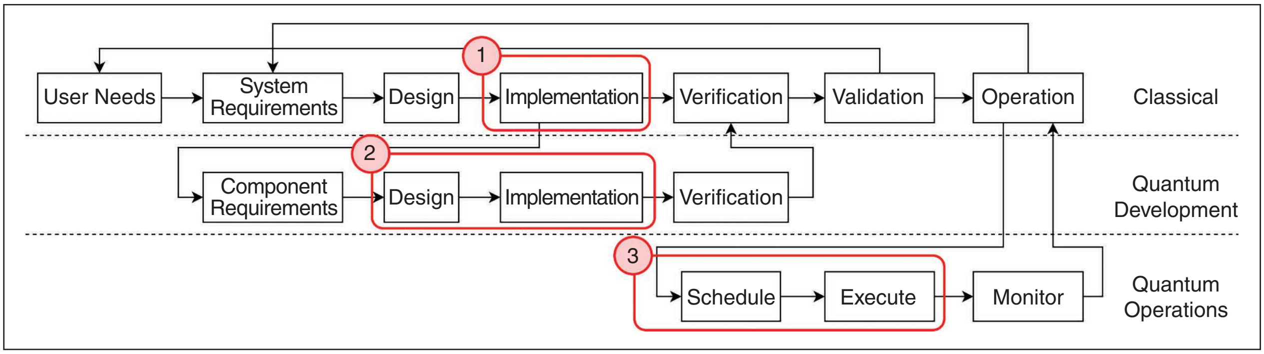

FIGURE 1. Quantum software development lifecycle and areas that contribute to gaining quantum advantage: 1) identify if use of quantum technology is feasible, 2) design and implement a quantum algorithm, and 3) find a quantum computer capable and available to execute the computation (adapted from Yue et al.[6])

QUANTUM COMPUTING GLOSSARY

- *Quantum annealing*: In quantum annealing the quantum system starts from an initial state and then evolves over time, gradually moving toward a state that represents the optimal solution to a given problem. Quantum annealing does not enable universal quantum computing, i.e., programming in any desired way, but it is suitable for many optimization tasks. While classical annealing is inherently probabilistic and relies on randomness to explore the solution space, quantum annealing exploits quantum effects, such as tunneling and entanglement, to traverse the solution space more efficiently.
- *The Ising model* is a mathematical model used in statistical mechanics to describe ferromagnetism in statistical physics.[9] The model consists of discrete variables that represent magnetic dipole moments of atomic spins that can be in one of two states (+1 or –1). The Hamiltonian for an Ising model describes the energy of the system in terms of the spins and their interactions.
- *Quadratic unconstrained binary optimization* (*QUBO*) is an Ising model with two possible states for spin can be directly mapped onto a QUBO problem. A QUBO problem is characterized by a goal of minimizing (or maximizing) a quadratic polynomial function of binary variables (0 or 1). The conversion from an Ising model to a QUBO model involves representing the Ising spins (+1, –1) as binary variables (0, 1) and then encoding the interactions and fields of the Ising model into a matrix of coefficients that define the QUBO problem. The diagonal entries of this matrix represent linear terms, while the off-diagonal entries represent quadratic interactions between variables.
- *Quantum approximate optimization algorithm* is a hybrid iterative method for solving combinatorial optimization problems. The method has been widely-studied method in the context of NISQ devices in particular.
- *Weight of evidence* (*WoE*)[10,11] is a statistical method used in credit scoring to evaluate variables' predictive power in distinguishing outcomes like defaulting. This process involves categorizing data into bins while ensuring no bins lack positive or negative cases. The calculation of WoE is based on computing the logarithm of the ratio of the distribution of good outcomes (nondefaults) to bad outcomes (defaults) within each bin. A higher WoE value indicates a stronger evidence that the variable is a good predictor of the outcome. However, despite its convenient property of handling missing values and outliers, it can result in information loss and may overlook variable correlations, underlining the importance of comprehensive data review before use.

SOME COMPLEX OPTIMIZATION AND STOCHASTIC MODELING PROBLEMS, SUCH AS PORTFOLIO OPTIMIZATION, DERIVATIVE PRICING, AND CREDIT SCORING

- *Portfolio optimization* is a problem where the goal is to choose the optimal combination of assets and their quantities from a given selection, based on a set of objectives that align with an investor's risk tolerance and expected return preferences. Modern portfolio theory explores the balance between risk and return, aiming at an efficient portfolio that maximizes expected returns for a specified level of risk.
- The framework for *pricing financial assets* is grounded in measure-theoretic probability theory. It models the market as a filtered probability space, incorporating potential market states, the flow of information over time, and a real-world market probability measure. In this framework, financial variables, like stock prices and interest rates, are modeled as stochastic processes. The market is assumed to be arbitrage-free, meaning it's impossible to make a riskless profit without investment. This assumption leads to the concept of an equivalent martingale measure (or risk-neutral measure), which ensures that a portfolio's current value matches its expected value at maturity, indicating no arbitrage opportunities and reflecting the fair value under this measure.
- *Credit scoring* is a task in finance, where the goal is to discern the independent features that significantly affect an applicant's creditworthiness. This selection of features, which must be both independent of each other and influential on the credit decision, can be formulated as a QUBO problem. The process involves starting with a dataset (matrix U) containing various features (columns) for past credit recipients (rows), and a vector representing past credit decisions. The objective is to select a subset of features that correlate highly with the creditworthiness decisions but have low intercorrelation. A quadratic objective function that balances these aspects is constructed, and quantum annealing can be applied to solve the QUBO, considering a penalty term to enforce the selection of a specific number of features. The selected features can be employed by machine learning algorithms, classical or quantum, to classify the creditworthiness of new applicants.

show promise on decreases in computation durations for some financial optimization tasks. QAE is used to estimate the probability of an outcome from a quantum measurement. In practical terms, it could be used to solve complex financial modeling problems, such as pricing derivatives. VQE, on the other hand, is an algorithm designed to find the lowest energy state, or the ground state of a quantum system and can be used to solve optimization problems in finance.

Quantum annealing has also demonstrated potential in a range of optimization scenarios. However, this paradigm does not constitute universal quantum computing, i.e., programming in any desired way, but it is suitable for many optimization tasks.

Our goal was to assess how close we are to achieving quantum advantage, where quantum computing outperforms traditional methods. We chose quantum annealing for its ability to tackle various optimization problems that could be useful for a financial institution. Inspired by a white paper on optimal feature selection,[8] we saw a potential application within the domain of credit risk management, who needed to refine financial models by identifying key features from a large set of data. The task was to select the top 10 features from 164 variables found in a credit information dataset with millions of rows, using quantum annealing techniques.

Quantum annealing can solve combinatorial binary optimization problems corresponding to an Ising model Hamiltonian, which has two possible states for spins.[9] This type of system can be represented as a QUBO problem by creating a matrix of coefficients that encodes the problem. In addition to the optimal feature selection problem, portfolio optimization is another problem from finance that can be formulated into a QUBO problem straightforwardly.

With the use of classical computer clusters, we calculated the needed correlation coefficient between features and creditworthiness after which these coefficients were then placed into a QUBO matrix, which corresponds to the optimization problem we wanted to solve. Formulating QUBO matrices is rather straightforward; however, depending on the data used to form the matrix, numerical problems might arise. To mitigate the complications arising from missing values in the data when performing correlation calculations, we chose to apply WoE transformation[10,11] into the original data. WoE is a statistical measure with a convenient property of handling missing values and outliers. It is used in various domains including finance, risk management, and machine learning, particularly in the development of credit scoring models.

Solving NP-hard problems using quantum annealing hardware typically involves a compilation (embedding) step, which is equivalent to the graph minor embedding problem,[12] which is itself an NP-hard problem. One step forward has been the development of heuristic algorithms,[13] which can find valid graph minor embeddings for a given problem instance in a reasonable amount of time.

In our case, the hardware topology was the limiting factor for fitting the problem into the machine in its entirety. Due to limited connectivity between qubits, a +5,000 qubit machine (D-Wave Advantage System 5.3[14]) was able to process only a QUBO with 66 variables for our specific problem. For the problem sizes we tried to solve with quantum annealing, the computational cost of finding graph minor embedding overshadows any performance gain that could potentially be achieved with the use of QA hardware when compared to simulated annealing.

Heuristic algorithms can find good enough solutions for some NP-hard problems (such as finding the minor embedding), and this has been our experience with using simulated annealing to solve QUBO problems.

The situation might be different in the case of sparse QUBO problems with more variables for which limited connectivity is not a concern, but in our case hardware topology does not allow us to perform optimization for QUBO with such a variable count that we could demonstrate any increase in solution quality over simulated annealing.

To summarize, in terms of the three quantum advantage factors described previously, the experiment provided the following new information, which correspond to the three areas of interest highlighted in Figure 1:

- *Partitioning the problem*: The task involved selecting the top 10 features from a large dataset for credit risk model. It is essential to discern the independent features that significantly affect an applicant's creditworthiness. Finding the ideal combination of features is an ideal candidate for quantum annealing because it is a combinatorial optimization problem. Classical computing was used to produce a QUBO matrix that corresponds to the optimization problem, which can be solved efficiently with quantum annealing.
- *Quantum algorithm*: When selecting a quantum algorithm for a quantum annealing, the decision is heavily influenced by the choice of hardware since commercial quantum annealing providers lock in specific, optimized algorithms tailored to their unique hardware design and control limitations, leading to a scenario where the hardware selection dictates the available annealing algorithms. We chose to use *DWaveSampler* as the quantum annealing algorithm and *SimulatedAnnealingSampler* for simulated annealing, both from DWave's Ocean library.[14] In our case, we only have a high abstraction level access to the quantum annealer through readily available application programming interface calls (https://docs.ocean.dwavesys.com/projects/system/en/stable/). The only way we can influence the annealing process is to search for the best annealing parameters that can be tweaked by the user.
- *Quantum processor limitations*: Despite using the D-Wave Advantage System 5.3 with more than 5,000 qubits, hardware limitations, specifically qubit connectivity, restricted the problem to a QUBO with 66 variables at most. This limitation prevented a full demonstration of quantum advantage over classical methods like simulated annealing, due to the advantage becoming more apparent as the number of variables in the problem increase. To give some context regarding runtimes, we compare execution times between the chosen simulated annealing and quantum annealing algorithms in a modest execution environment (Apple M2 laptop for simulated annealing). If we consider only the annealing times, quantum annealing is faster by a factor of 10^6, but the wall time of quantum annealing is about 500 times slower

compared to simulated annealing when pre- and postprocessing steps are considered. This slowdown is mostly due to the requirement of first finding a minor embedding to map the QUBO into the quantum processor (QPU) topology, which is an expensive operation, although it needs to be done only once for a single QUBO matrix. If execution time for finding the minor embedding is not considered, quantum annealing was about 10 times faster compared to simulated annealing with modest hardware. When comparing solution quality between quantum annealing and simulated annealing, we saw 21% difference in minimum energies in favor of simulated annealing when the number of selected variables was controlled to be the top two variables and 6% difference in favor of simulated annealing when selecting top three variables. From four variables up to eight variables, the solutions and minimum energies were identical between the two methods.

In the context of quantum software development, observability and debugging tools like the dwave-inspector[14] are crucial for understanding physical qubit chains representing logical variables in a particular form of quantum annealing. To ensure accurate solutions, all qubits representing a single logical qubit, called a qubit *chain*, should ideally exhibit uniform values postannealing. The qubit chains are constrained to have the same value, 0 or 1, by a single parameter, called *chain strength*, which is an essential characteristic for the accuracy of quantum computations. If the chain breaks, a discrepancy between the qubit chain values arises; algorithms such as majority vote to are used to deduce correct variable assignments. However, the chain strength should be set to such a value that chain breaks don't happen in the first place to ensure that the solution is valid for the original problem.[14] Identifying and rectifying chain strength issues with the help of proper tooling is crucial for getting reliable results with quantum annealing.

Formal verification methods can be applied to quantum computing,[15] and in our case specifically in the construction of a correct QUBO matrix for optimal variable selection problem. These methods involve mathematical proofs or algorithmic checks that systematically verify the correctness of the QUBO matrix against the expected properties it should have (given a certain problem's specification). In our case. the verification of QUBO matrix construction with mathematical proof involves a simple evaluation of the objective function to show that the matrices we generate produce all of the correct terms of the known correct objective function for an arbitrary number of variables. Hence, formal verification methods can help developers to ensure that the problems solved by quantum computers are correctly formulated.

To effectively adapt algorithms for quantum computing, organizations need to focus research and development on converting classical algorithms to quantum versions. Unlike the straightforward lift-and-shift approach of cloud migration, not all problems are suitable for quantum computing. Thus, selecting workloads for quantum transition should concentrate on domains where quantum computing has distinct benefits, like optimization challenges in finance.

To effectively adapt algorithms for quantum computing, organizations need to focus research and development on converting classical algorithms to quantum versions.

Quantum computing has recently been gaining a lot of interest in the field of software engineering research. Topics—such as programming languages and, more generally, tools for composing quantum programs and deployment infrastructure to run quantum code—have been addressed. At the same time, practitioners have to work with NISQ era computers, maybe for decades. With them, the prime interest is to fully use these capabilities, such as easy to use optimized circuit routing based on qubit quality information, and they have, instead of chasing architectural abstractions that make sense in the fault tolerant quantum computing era.

ABOUT THE AUTHORS

PETRI LIIMATTA is a technology exploration lead with the OP Financial Group, 00510 Helsinki, Finland. Contact him at petri.liima@op.fi.

TEIKO HEINOSAARI is a full professor at the University of Jyväskylä, 40014 Jyväskylä, Finland. Contact him at teiko.heinosaari@jyu.fi.

PAULI TAIPALE is a technology strategist with the OP Financial Group, 00510 Helsinki, Finland. Contact him at pauli.taipale@op.fi.

TOMMI MIKKONEN is a full professor at the University of Jyväskylä, 40014 Jyväskylä, Finland. Contact him at tommi.j.mikkonen@jyu.fi.

KIMMO HALUNEN is a full professor at the University of Oulu, 90570 Oulu, Finland. Contact him at kimmo.halunen@oulu.fi.

VLAD STIRBU is a postdoctoral researcher at the University of Jyväskylä, 40014 Jyväskylä, Finland. Contact him at vlad.a.stirbu@jyu.fi.

Moreover, available quantum infrastructure and hardware leaves many industrial organizations unsatisfied, as the facilities presently available fail to meet the expectations and the scale of practical problems. Rather, in the advent of more capable quantum computers, the question is how to partition these problems so that quantum advantage can be gained with particular, specialized subproblems that are small enough. Even with such, simulators might be a better solution because of their better computational capacity.

Hence, a key challenge of software engineering research is to predict timely investments in the quantum technology, related software, applicable development methods, as well as industry-scale applications. This calls for open-minded real-world prototypes, where the limits of the technology are probed in practice. Hence, we should experiment more to learn about the engineering principles of quantum software, instead of recirculating classical software engineering results in the quantum context without exploratory goals or connection to practical industry needs.

References

1. F. Gaitan, "Finding flows of a Navier–Stokes fluid through quantum computing," *npj Quantum Inf.*, vol. 6, no. 1, 2020, Art. no. 61, doi: 10.1038/s41534-020-00291-0.
2. T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E*, vol. 58, no. 5, pp. 5355–5363, 1998, doi: 10.1103/PhysRevE.58.5355.
3. J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, Aug. 2018, Art. no. 79, doi: 10.22331/q-2018-08-06-79.
4. B. Weder, J. Barzen, F. Leymann, M. Salm, and D. Vietz, "The quantum software lifecycle," in *Proc. 1st ACM SIGSOFT Int. Workshop Archit. Paradigms Eng. Quantum Softw., Co-Located ESEC/FSE*, 2020, pp. 2–9, doi: 10.1145/3412451.3428497.
5. A. A. Khan, D. Taibi, C. M. Perrault, and A. A. Khan, "Advancing quantum software engineering: A vision of hybrid full-stack iterative model," 2024, *arXiv:2403.11670*.
6. T. Yue, W. Mauerer, S. Ali, and D. Taibi, "Challenges and opportunities in quantum software architecture," in *Software Architecture*, P. Pelliccione, R. Kazman, I. Weber, A. Liu, Eds., Cham, Switzerland: Springer-Verlag, 2023, pp. 1–23, doi: 10.1007/978-3-031-36847-9_1.

7. K. Bharti et al., "Noisy intermediate-scale quantum algorithms," *Rev. Modern Phys.*, vol. 94, no. 1, 2022, Art. no. 015004, doi: 10.1103/RevModPhys.94.015004.
8. A. Milne, M. Rounds, and P. Goddard, "Optimal feature selection in credit scoring and classification using a quantum annealer," Vancouver, BC, Canada, White Paper, 2017. [Online]. Available: https://1qbit.com/whitepaper/optimal-feature-selection-in-credit-scoring-classification-using-quantum-annealer/
9. B. A. Cipra, "An introduction to the Ising model," *Amer. Math. Monthly*, vol. 94, no. 10, pp. 937–959, 1987, doi: 10.1080/00029890.1987.12000742.
10. I. J. Good, *Probability and the Weighing of Evidence*. New York, NY, USA: Griffin, London and Hafner Publ., 1950.
11. N. Siddiqi, *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. New York, NY, USA: Wiley, 2006.
12. L. Lovász, "Graph minor theory," *Bull. Amer. Math. Soc.*, vol. 43, no. 1, pp. 75–86, 2006, doi: 10.1090/S0273-0979-05-01088-8.
13. J. Cai, B. Macready, and A. Roy, "A practical heuristic for finding graph minors," 2014, *arXiv:1406.2741*.
14. D-Wave, "Programming the D-wave QPU: Setting the chain strength," Burnaby, BC, Canada, White Paper, 2020. [Online]. Available: https://www.dwavesys.com/media/vsufwv1d/14-1041a-a_setting_the_chain_strength.pdf and at https://github.com/dwavesystems/dwave-ocean-sdk
15. M. Lewis, S. Soudjani, and P. Zuliani, "Formal verification of quantum programs: Theory, tools, and challenges," *ACM Trans. Quantum Comput.*, vol. 5, no. 1, 2023, Art. no. 1, doi: 10.1145/3624483.

Digital Object Identifier 10.1109/MS.2024.3459106