A Bayesian Network-based Quantum procedure for Failure Risk Analysis

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

22-12-2021 / 28-12-2021

CITATION

Carrascal de las Heras, Ginés; Botella, Guillermo; Del Barrio, Alberto A; Kremer, David (2021): A Bayesian Network-based Quantum procedure for Failure Risk Analysis. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.17402915.v1>

DOI

[10.36227/techrxiv.17402915.v1](https://doi.org/10.36227/techrxiv.17402915.v1)

A Bayesian Network-based Quantum procedure for Failure Risk Analysis

Ginés Carrascal^{1,2}, Guillermo Botella³, Alberto del Barrio³, and David Kremer¹

¹IBM Consulting España, 28830 Madrid, SPAIN

²Department of Informatic Systems and Computation, Faculty of Informatics, Complutense University of Madrid, 28040 Madrid, SPAIN

³Department of Computer Architecture and Automation, Faculty of Informatics, Complutense University of Madrid, 28040 Madrid, SPAIN

Corresponding author: Ginés Carrascal (email: ginescar@ucm.es).

ABSTRACT Studying the propagation of failure probabilities in interconnected systems like that of electrical distribution networks is traditionally performed by means of Monte Carlo simulations. In this paper, we propose a procedure to create a model of the system in a quantum computer using a restricted representation of Bayesian networks. Some examples of this implementation on sample models are presented using Qiskit and tested using both quantum simulators and IBM Quantum hardware. Results show a correlation in the precision of the results when considering the number of Monte Carlo iterations alongside the sum of shots in a single quantum circuit execution.

INDEX TERMS Bayesian Network, Quantum computing, Risk analysis, Resilience analysis

I. INTRODUCTION

Efficient electricity distribution networks compose a crucial component of modern society today. Therefore, modelling such systems for the purpose of mitigating against possible failures and power outages is an important area of investigation. This type of study is referred to as reliability analysis. Formally, we define the reliability of a system as the probability that the system in question will operate or perform a certain function, under fixed conditions and for a specified period of time. In this study, we consider electrical substations as one of the most important components of an electric power distribution network. However, combinations of unit element failures in these networks can result in a critical loss of load. Therefore, calculating the most likely modes of failure, or those involving fewer individual elements can help in planning preventive maintenance. Using aging models in combination with data from sensors on the elements themselves, we can calculate the probability of failure of each of the individual elements of the distribution network. This study will calculate the probability that the electrical substations will remain in operation based on the joint probabilities of failure of each of their individual components:

transformers, bars, switches, lines, as well as the protection systems themselves.

To work with conditional probabilities of this type we use Bayesian network models, which are commonly used in this type of research. The difficulty associated with these models is their high computational complexity. As the size of the problem increases, where in the case of Bayesian networks this is measured by the number of nodes and arcs needed to model the problem, the time and computational memory of the classical algorithms to solve these models grows exponentially, to a point that their resolution is classically unfeasible.

Gate-based quantum computers are expected to help solve problems in quantum chemistry [1][2][3], machine learning [4][5], financial simulation [6][7][8][9] and combinatorial optimization application areas [10][11].

As predicted by Preskill [12], Noisy Intermediate-Scale Quantum (NISQ) computers with more than 100 qubits are now a reality and may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. To obtain all the advantages of this

technology, we will need more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

In this paper, we evaluate whether this technology will also be able to help in resilience and failure risk analysis.

We highlight the new contributions in this work:

- Define a new Restricted Quantum Bayesian Network procedure to model the resilience of complex systems.
- Evaluate the viability of the procedure to perform resilience analysis, obtaining the same accuracy as classical Montecarlo by adjusting the number of shots in a single quantum circuit execution.
- Test the impact of real quantum computers noise in the elements of the model.

The rest of the paper is organized as follows: Section II defines the objectives. Section III recaps the foundations needed by the Restricted Quantum Bayesian Network model described in Section IV. Sections V and VI explore two case studies for the defined procedure. Section VII deals with the implications of noise in real quantum computers. And section VIII summarizes the conclusions.

II. OBJECTIVES

This paper proposes a procedure for the general application of quantum modelling for the propagation of probabilities of failure along an electrical distribution system, as well as a procedure for the calculation of this model.

Several electrical power distribution systems (in particular, a typical electrical substations and a protection system) will be modelled by Bayesian networks as examples. The failure modes of the substations will be calculated with both classical algorithms and quantum algorithms.

We approach solving Bayesian networks in a classical way using *pomegranate*, a Python library that is able to implement probabilistic models [13]. We then look to model and solve Bayesian networks in a quantum regime using Qiskit [14], due to its lower code effort [15].

This approach allows us to validate the results and analyze the performance of both solutions.

III. FOUNDATIONS

A. RESILIENCE ANALYSIS

The resilience of a system is related to its ability to withstand stress, adapt to this stress, and avoid disruptions.

Electrical substations resilience is approached classically with different methods by many authors, including fault tree and event tree analysis [16] or failure modes and effects evaluation [17], and also Bayesian networks [18], and with this, the reliability of the system can be obtained using probability propagation techniques. This allows for modelling complex systems, such as a bridge type, and dependencies between

failures, which are difficult to obtain with conventional reliability analysis techniques.

B. BAYESIAN NETWORKS

A Bayesian network is a probabilistic graphical model used to represent situations with associated uncertainties, where nodes represent random variables (discrete or continuous) and arcs connecting pairs of nodes represent connections (dependencies) between the nodes. Figure 1 shows a simple network and the elements that compose it.

FIGURE 1. Example of a Bayesian Network ($Z = N_1 \wedge N_2$).

The only restriction on arcs connecting nodes in a Bayesian network is that there must be no cyclic relationships, that is, you cannot return to a node simply by following a series of arcs. Such networks are called directed acyclic graphs (DAGs).

The function of Bayesian networks is to calculate the *a posteriori* distribution of a set of nodes, given the values of other nodes which constitute evidence or observations. This process is called updating probabilities, or beliefs, and is much like representing a flow of information through the network.

To make such updates, Bayesian networks require the iterative application of Bayes' Theorem, and for this purpose exact or approximate inference algorithms can be used.

There are some software programs that use such algorithms and make it possible to build and use Bayesian networks without knowing all the details of the underlying probability update algorithms. Specifically, in this study we will use a Python library called *pomegranate* [19], that uses either maximum-likelihood estimates or expectation-maximization. For larger networks, the use of an exact algorithm may become unfeasible, and in that case approximate algorithms should be used. In general, both exact and approximate algorithms theoretically have exponential complexity and are considered to be NP hard [20].

In practice, the speed of the algorithm will depend on factors such as the structure of the network, the number of connections, or the location where the observation nodes are located.

C. QUANTUM GATES

To be able to model these conditional probabilities, we will need the following Quantum Gates:

X gate

The quantum X gate is the equivalent of the classical NOT gate. This swaps, or flips, basis states and is represented as:

$$X = U_3\left(\pi, -\frac{\pi}{2}, \frac{\pi}{2}\right) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

R_Y gate

This gate corresponds to a rotation of θ radians around the y-axis of the Bloch sphere. It is represented as:

$$R_Y(\theta) = U_3(\theta, 0, 0) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Two-qubit quantum gates

The elementary gate composed of two qubits is the controlled-NOT (CNOT or CX) gate. The qubits on which this gate is implemented are known as control and target qubits.

When the control qubit has value $|0\rangle$, the target qubit does not change its state, whereas when the control qubit has value $|1\rangle$, the X gate is applied over the objective qubit to change its state. In the computational basis with states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, the matrix representing the CNOT gate is as follows:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Another gate similar to the CNOT gate is the $CR_Y(\theta)$ gate, which implements a rotation $R_Y(\theta)$ when the control qubit has the value of $|1\rangle$. It is represented as:

$$CR_Y(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ 0 & 0 & \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Three qubit and other quantum gates

The two three-qubit gates we use for our algorithm are the CCNOT (CCX or Toffoli) and the $CCR_Y(\theta)$. In these gates, two qubits act as control qubits and the final qubit in the register acts as the target qubit. In the case of the CCNOT gate, when the two control qubits have the value $|1\rangle$, the X gate is

applied to the target qubit. In the case of the $CCR_Y(\theta)$ gate the $R_Y(\theta)$ is applied. Thus, we can now define the gates C^nX and $C^nR_Y(\theta)$ for n control qubits and a target qubit.

Expansion of controlled gates

To be able to model conditional probabilities, we will need to perform controlled rotations in our quantum circuits. This controlled rotation can be efficiently mapped to simple, one qubit, rotations and CNOT gates.

We can use the following expansion equivalence to implement controlled rotation gates (Figure 2)

FIGURE 2. Expansion equivalent to controlled gate.

IV. RESTRICTED QUANTUM BAYESIAN NETWORK MODEL

In a general way, Sima E. Borujeni et al. had established a method to map conditional probabilities from a Bayesian Network into a Quantum Circuit [21], resulting in high depth circuits with 2^n operations per node, with n being the number of parent nodes of the node in question.

To model the probability of failure of the electrical grid, we have a particular case of the Bayesian network, in which the leaf nodes will be the random variables representing the failure probability of each individual element. The rest of the nodes will be modeled with combinations of “AND” and “OR” nodes, depending on the serial or parallel configuration of the elements respectively.

Under these circumstances, we define what we call a Restricted Quantum Bayesian Network (RQBN) that leads to a low depth circuit, with only one operation per node. This formation is more suitable to run in currently available NISQ hardware.

To solve this problem in the quantum regime, we follow these steps to design a quantum circuit that represents our Bayesian network:

1. Assign each node of our Bayesian network to a qubit. All the nodes that we are going to discuss in this paper have only two states (0 if the element works, 1 if it fails). Since qubits can represent two states, we can assign each node to a different qubit.
2. Assign the marginal or conditional probabilities of each node to the probability amplitudes associated with the states of the qubits.
3. Obtain the desired probability amplitudes of qubit states using controlled rotation gates.

A. ROOT NODES

To represent a root node with two states we will apply a $R_Y(\theta)$ gate with an appropriate angle θ . Thus, the probabilities of the

$$\theta_{V_i} = 2 \times \tan^{-1} \sqrt{\frac{P(|1\rangle)}{P(|0\rangle)}} = 2 \times \tan^{-1} \sqrt{\frac{P(V_i = 1)}{P(V_i = 0)}}$$

root nodes can be assigned to the probabilities (and therefore to the probability amplitudes) of the basic states $|0\rangle$ and $|1\rangle$. Let θ_{V_i} be the angle representing the angle of rotation associated with a two-state node V_i . If the initial state of a qubit is $|0\rangle$, $R_Y(\theta)$ transforms $|0\rangle$ to $\cos(\theta/2)|0\rangle + \sin(\theta/2)|1\rangle$. Therefore, the probabilities associated with the states $|0\rangle$ and $|1\rangle$ are $\cos^2(\theta/2)$ and $\sin^2(\theta/2)$ respectively. If $P(V_i = 0)$ and $P(V_i = 1)$ represent the probabilities of the states 0 and 1 of V_i , then we can calculate the rotation angle for the node V_i as:

B. "AND" NODE FOR SERIAL ELEMENTS

A serial system like the one in Figure 3 can be modeled with a Bayesian network like the one in Figure 1:

FIGURE 3. Simple serial system.

Nodes N_1 and N_2 refer to elements 1 and 2 of the system in series while node Z corresponds to the probability that the entire system will continue to run. Each element that is part of the system has a probability of operation of $P(N_i) = p_i$ and a probability of failure of $P(\neg N_i) = q_i$.

These probabilities are estimated by domain experts and are calculated by considering several influential factors such as the type of element involved, the duration of time it has been running, or if it has had any previous repair.

On the other hand, the probability of operation of the Z node (i.e. of the complete system) is conditioned by nodes N_1 and N_2 and is quantified using the conditioned probability table in Table I.

TABLE I
CPT FOR SERIAL SYSTEM. NODE TYPE "AND"

	(0,0)	(0,1)	(1,0)	(1,1)
$P(Z=0 N_1, N_2)$	1	1	1	1
$P(Z=1 N_1, N_2)$	0	0	0	0

In a serial system, we need the operation of both elements for the system to be successful, so we can use the Conditional Probability Table (CPT) of node Z as a truth table of a logical conjunction. Thus, the reliability of this system (i.e., the probability that the system will work) is:

$$P(z) = 1 \cdot p_1 p_2 + 0 \cdot q_1 p_2 + 0 \cdot p_1 q_2 + 0 \cdot q_1 q_2$$

And the probability of system failure is:

$$P(z) = 0 \cdot p_1 p_2 + 1 \cdot q_1 p_2 + 1 \cdot p_1 q_2 + 1 \cdot q_1 q_2$$

We can implement this behavior as a controlled rotation of π radians and use the expansion of controlled gates to obtain the circuit in Figure 4.

FIGURE 4. "AND" node for 2 control qubits.

C. "OR" NODE FOR PARALLEL ELEMENTS

Similarly, we can explain how to implement a parallel system such as the one represented in Figure 5.

FIGURE 5. Simple parallel system.

The same Bayesian network of the serial system can be used in this case. The only difference is that for the parallel system to continue functioning successfully, it needs only one of the elements (1 or 2) to be functional. Therefore, the CPT of node Z acts as the truth table of a logical disjunction, as in Table II.

TABLE II
CPT FOR PARALLEL SYSTEM. NODE TYPE "OR"

	(0,0)	(0,1)	(1,0)	(1,1)
$P(Z=0 N_1, N_2)$	1	0	0	0
$P(Z=1 N_1, N_2)$	0	1	1	1

We can implement this behavior as an expansion in a similar way as previously done for the "AND" node, but in this case substituting the Toffoli gates with a circuit implementing the OR logic, as depicted in Figure 6.

FIGURE 6. "OR" node for 4 control qubits.

This implementation is scalable to any number of control qubits, representing nodes in the Bayesian Network connected to the node in question.

D. FINDING MOST COMMON FAILURE MODES

Using this representation we can obtain the probability of each combination of failure scenarios of the root elements, as well as the corresponding state of the combination nodes, including the final node that represents the failure of the complete circuit. Thus, it is straightforward to obtain all measures with a probability of 1 for the complete circuit, and the cause of the failure, simply by observing the results corresponding to each individual element.

V. CASE STUDY: SUBSTATION FAILURE

A. SINGLE BUS SUBSTATION

A single bus substation is the simplest configuration for an electrical substation.

FIGURE 7. Single bus substation
(Source: Electrical Engineering Portal [22]).

This consists of four breakers and a bar that transfers the energy from the lines to the transformers, as shown in Figure 7.

We can perform a path analysis to create a Bayesian Network representing this substation configuration [23].

Each individual element in the substation will be represented by a root node, with the associated failure probability of this element. Items labelled 1 and 2 are the lines in this case, 3,4,5 and 6 represent the breakers, 7 and 8 the transformers and 9 the bus. In a real scenario, it is possible to assign an individualized failure probability to each element, based on the nature of the element, its associated failure and repair history, environmental conditions, sensor measurements amongst other factors.

We can determine the different available paths for the electricity flow. These paths will be represented by “AND” nodes, because they are formed by elements in series.

To model the complete system status, all the paths have to be combined in an “OR” node, representing the case where all these paths are possible parallel circuits for the electricity flow. The resulting Bayesian Network is represented in Figure 8:

FIGURE 8. Bayesian network for a single bus substation
(Source: Łukasz Wojdowski [23]).

FIGURE 9. Quantum circuit for single bus substation.

With this approach we can model more complex real-life scenario circuits. We have selected this example for ease of explanation.

B. RESULTS

We have modeled the aforementioned Bayesian network classically using the open-source library *pomegranate* and using Qiskit for our quantum circuit.

The resulting circuit is shown in Figure 9, and the comparison of the probabilities of each mode obtained classically and in the quantum regime are shown in Figure 10. In Figure 10 the results ending in "1" correspond to "working modes", and the results ending in "0" correspond to "failure modes".

Thus, if we only consider the results ending in "0", it is possible to compare the probability of failure of each combination of failing elements, as shown in Figure 11.

FIGURE 10. Bayesian network results for a single bus substation (red classical, blue quantum).

FIGURE 11. Failure modes for a single bus substation (red classical, blue quantum).

VI. CASE STUDY: PROTECTION SYSTEMS

A. PROBABILITY OF FAILURE OF A PROTECTION SYSTEM

A substation bus is a unit of aggregation in the electrical grid that consists of a modular set of elements connected at the same voltage level and constitutes the unit of operation of the network. A bus is typically protected by a dedicated system.

When a fault is detected in any of the elements connected to the bus, this protection system closes the position, preventing the fault from propagating further through the network. A failure in the protection system can therefore result in a large-scale network outage.

A protection system is formed by a set of "protections", digital or analogue units that perform different monitoring functions. Typically, each protection unit can perform a set of functions, and different units are combined to provide the complete set of protection functions required by the bay, usually in a redundant way. If a protection unit fails, the specific protection functions it performs are lost.

The purpose of this exercise is to estimate the probabilities of failure of the protection system, that is, the probability of the system not fulfilling its purpose for different combinations of failures of individual protection units based on the given probabilities of failures of the individual units of the system. Specifically, we search for the most probable combination of unit failures that leads to a system failure, and for the combination that involves the minimum number of units experiencing failure that then led to a system failure.

For the purpose of this exercise, these functions were classified in "essential" and "non-essential" functions. With this classification, a protection system is said to fail if and only if:

1. ONE essential function is lost. That is, if any of the essential functions are not safeguarded by any of the working protection units.
2. ALL non-essential functions are lost. That is, if there are no working units that are able to provide non-essential functions.

For example, if a protection system requires functions A, B and C (essential) and D and E (non-essential), the system will fail if all the units safeguarding A fail (or B or C), or if all units providing D and all units providing E fail. For each protection system we identify each of the essential and non-essential functions with a sequential ID.

For each of the protection units, the functions they provide are stored in two vectors that encode their essential and non-essential functions in terms of the set of functions of the system. For example: if the system has 3 essential and 2 non-essential functions, a protection unit characterized by $[1,0,1],[0,1]$ will imply that it performs essential functions 0 and 2 and non-essential function 1.

A function for generating a random dataset for a protection system was created. The number of units, essential functions and non-essential functions are configurable.

B. SOLUTION APPROACH 1: EXACT CALCULATION

A straightforward way to compute the probability of failure of the system is to enumerate all combinations of unit failures, compute the probability of each of these and then determine, for each unit, if they result in a system failure based on the failure rules described earlier.

For this, we first need a protocol that, given a combination of failed units, the functions provided by these units, and the necessary functions at the system level, determines if this combination results in a system failure.

Then we need a protocol that generates all possible samples, or combinations of failures, given a number of units. Note that the number of samples grows exponentially with the number of units (2^n).

We also need a function that computes the probability of a sample given the probability of failure of each individual unit. While the exact calculation is possible for small systems, it becomes unfeasible for larger ones. In fact, exact calculations for more than 20 units and 10 functions are considered computationally expensive in modern hardware, both in terms of memory and CPU time. On the other hand, systems with more than 20 units are not uncommon in practice, and they may even reach numbers larger than 50.

C. SOLUTION APPROACH 2: MONTECARLO ESTIMATION

In this Monte Carlo approach, instead of considering all possible samples, random samples are generated according to their probability of occurrence. As in the exact method, we then determine if each of the generated samples result in a system failure based on the aforementioned failure rules. Finally, the system failure probability is calculated as the proportion of resulting failures over the total number of samples.

In this approach, the number of samples is parameterized. More samples will mean a better estimation of the system probability. In Figure 12, we can see a comparison of execution time calculating the overall failure probability for the system for a range of units from 2 to 20 using the exact approach and the Monte Carlo approach with a fixed number of 2^{18} iterations (Executions on a 2,4 GHz 8-Core Intel Core i9; 32 GB 2667 MHz DDR4).

FIGURE 12. Comparison of overall failure calculation time exact vs Montecarlo.

It must be noted that, in a more realistic scenario, to maintain the precision of the Monte Carlo method, we should increase the number of iterations as the number of units increase.

D. SOLUTION APPROACH 3: QUANTUM BAYESIAN NETWORKS

We can model the behavior of the protection systems with a Restricted Quantum Bayesian Network. The root nodes will represent the individual units with a R_Y gate rotating the qubit in order to represent the failure probability of the corresponding unit.

A first layer of AND nodes will aggregate each of the protection functions. In a second layer, one AND node will aggregate the non-essential functions and an OR node the essential ones. Each root node will be connected only to those of these nodes representing the functions protected by this unit.

At the top layer, a last AND node will aggregate the previous two nodes, representing the overall status of the system.

Figure 13 represents an example Bayesian Network for 10 units protecting for 4 non-essential functions and 7 essential ones.

FIGURE 13. Bayesian Network for 10 units protecting for 4 non-essential functions and 7 essential ones.

Figure 14 shows a representation of the quantum circuit developed from the network in Figure 13. The exact results have not been represented to reduce the size of the image.

To compare with a Monte Carlo approach, a random system with 16 units protecting 7 essential functions and 4 non-essential functions was also generated. A random failure probability was also assigned to each unit.

For this comparison, only the overall failure probability of the complete system was calculated in an exact way, with both the

FIGURE 14. Quantum circuit for 10 units protecting 4 non-essential functions and 7 essential ones.

Monte Carlo and with the Restricted Quantum Bayesian Network (RQBN). Figure 15 shows the mean error compared with the exact calculation for both Monte Carlo and RQBN approach.

FIGURE 15. Comparison of overall failure calculation error Montecarlo vs RQBN.

It is important to note that in Figure 15, the X axis represents the number of iterations for the Monte Carlo algorithm but also the number of shots in an individual execution of the RQBN circuit. So, we can adjust the desired accuracy of our quantum method with the number of shots to match the corresponding accuracy of a Monte Carlo method with a defined number of iterations.

VII. EXECUTION ON REAL QUANTUM COMPUTERS

A. LIMITED NUMBER OF QUIBITS

To validate the feasibility of the RQBN procedure on NISQ real quantum computers, and to simplify the comparison of experimental results, we will evaluate the Bayesian network shown in Figure 8 in several Quantum computers from the IBM Quantum cloud services¹.

B. NOISE IMPLICATIONS

In NISQ computers noise accumulates with the depth of the circuits and affects the capability of using the theoretical algorithms speed up to solve realistic problems.

Noise provides a natural constraint on accessible circuit depths and is scales exponentially to penalize larger circuit depths, leading to an exponential decoherence of quantum states.

As a result, more shots are needed to battle the noisy information.

Of course, we need not forget that equivalent quantum circuits for the Bayesian Networks may already result in better quality quantum computers than the ones available today, so observing these quantum speedups in practice is not yet possible.

C. Experimental results

We have tested a simple Bayesian Network consisting of only one “AND” node in two different real quantum computers “ibmq_quito”, with 5 qubits and Quantum Volume 16 (CNOT average error $1.337e-2$). and “ibmq_manila”, also with 5 qubits but Quantum Volume 32 (CNOT average error $7.673e-3$).

In both cases the measures are taken with 8182 shots.

The results are shown in Figure 16 and Figure 17.

FIGURE 16. Bayesian network results for a single “AND” node (red classical, blue real quantum computer “ibmq_quito”).

¹<https://quantum-computing.ibm.com/services?services=systems&systems=yours>

FIGURE 17. Bayesian network results for a single “AND” node (red classical, blue real quantum computer “ibmq_manila”).

The results for “ibmquito” are slightly less accurate due to a larger error rate. This is more easily observed in the “111” result, as it has the lower probability and so is more easily mistaken than with results like those from errors associated with “110” or “001”.

VIII. CONCLUSIONS

It is possible to represent the conditional probabilities of failure of an electrical distribution network in an efficient way in a quantum computer. For this we have proposed a Restricted Quantum Bayesian Network (RQBN) procedure that leads to substantially shorter circuits than a general representation.

We have implemented this model for different scenarios and validated the viability of such representation. Comparing with Monte Carlo methods, the results obtained are similar in accuracy for the same number of shots in a single quantum circuit execution than with the number of iterations of the classical Monte Carlo simulation.

Finally, viability and implications of noise in real quantum computers has been tested with promising results.

ACKNOWLEDGEMENTS

The authors acknowledge useful discussions with Kate Marshall, Armand Stekelenbur, Hedayat Alghassi and Nicolas Robles. IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at <https://www.ibm.com/legal/copytrade>

REFERENCES

- [1] N. Moll *et al.*, “Quantum optimization using variational algorithms on near-term quantum devices,” *Quantum Sci. Technol.*, vol. 3, no. 3, p. 030503, Jul. 2018, doi: 10.1088/2058-9565/aab822.
- [2] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, “Error mitigation extends the computational reach of a noisy quantum processor,” *Nature*, vol. 567, no. 7749, pp. 491–495, Mar. 2019, doi: 10.1038/s41586-019-1040-7.
- [3] M. Ganzhorn *et al.*, “Gate-Efficient Simulation of Molecular Eigenstates on a Quantum Computer,” *Phys. Rev. Applied*, vol. 11, no. 4, p. 044092, Apr. 2019, doi: 10.1103/PhysRevApplied.11.044092.
- [4] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017, doi: 10.1038/nature23474.
- [5] V. Havlíček *et al.*, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019, doi: 10.1038/s41586-019-0980-2.
- [6] D. J. Egger *et al.*, “Quantum Computing for Finance: State-of-the-Art and Future Prospects,” *IEEE Trans. Quantum Eng.*, vol. 1, pp. 1–24, 2020, doi: 10.1109/TQE.2020.3030314.
- [7] S. Woerner and D. J. Egger, “Quantum risk analysis,” *npj Quantum Inf.*, vol. 5, no. 1, p. 15, Dec. 2019, doi: 10.1038/s41534-019-0130-6.
- [8] P. Rebentrost, B. Gupta, and T. R. Bromley, “Quantum computational finance: Monte Carlo pricing of financial derivatives,” *Phys. Rev. A*, vol. 98, no. 2, p. 022321, Aug. 2018, doi: 10.1103/PhysRevA.98.022321.
- [9] D. J. Egger, R. Garcia Gutierrez, J. Cahue Mestre, and S. Woerner, “Credit Risk Analysis using Quantum Computers,” *IEEE Trans. Comput.*, pp. 1–1, 2021, doi: 10.1109/TC.2020.3038063.
- [10] L. Braine, D. J. Egger, J. Glick, and S. Woerner, “Quantum Algorithms for Mixed Binary Optimization Applied to Transaction Settlement,” *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–8, 2021, doi: 10.1109/TQE.2021.3063635.
- [11] P. Kl. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, “Improving Variational Quantum Optimization using CVaR,” *Quantum*, vol. 4, p. 256, Apr. 2020, doi: 10.22331/q-2020-04-20-256.
- [12] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018, doi: 10.22331/q-2018-08-06-79.
- [13] “Home — pomegranate 0.13.2 documentation.” <https://pomegranate.readthedocs.io/en/latest/> (accessed Jul. 11, 2021).
- [14] H. Abraham *et al.*, *Qiskit: An Open-source Framework for Quantum Computing*. 2019. doi: 10.5281/zenodo.2562110.
- [15] G. Carrascal, A. A. del Barrio, and G. Botella, “First experiences of teaching quantum computing,” *J Supercomput.*, vol. 77, no. 3, pp. 2770–2799, Mar. 2021, doi: 10.1007/s11227-020-03376-x.
- [16] F. Wang, B. W. Tuinema, M. Gibescu, and M. van der Meijden, “Reliability evaluation of substations subject to protection system failures,” in *Proceedings - 2013 IEEE Grenoble PowerTech*, 2013, pp. 1–6. doi: 10.1109/PTC.2013.6652162.
- [17] D. Nack, “Reliability of Substation Configurations,” p. 19.
- [18] J. Gerardo, T. Toledano, C. M. México, L. Enrique, and S. Succar, “Bayesian Networks for Reliability Analysis of Complex Systems.”
- [19] J. Schreiber, “Pomegranate: fast and flexible probabilistic modeling in python,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5992–5997, 2017.
- [20] D. M. Chickering, D. Heckerman, and C. Meek, “Large-Sample Learning of Bayesian Networks is NP-Hard,” *Journal of Machine Learning Research*, vol. 5, no. Oct, pp. 1287–1330, 2004.
- [21] S. E. Borujeni, S. Nannapaneni, N. H. Nguyen, E. C. Behrman, and J. E. Steck, “Quantum circuit representation of Bayesian networks,” *Expert Systems with Applications*, vol. 176, p. 114768, Aug. 2021, doi: 10.1016/j.eswa.2021.114768.
- [22] E.-E. E. Portal, “6 common bus configurations in substations up to 345 kV | EEP,” *EEP - Electrical Engineering Portal*, Mar. 18, 2019. <https://electrical-engineering-portal.com/bus-configurations-substations-345-kv> (accessed Dec. 19, 2021).
- [23] Ł. Wojdowski, “Substation Reliability Evaluation with Dependent Outages and Switching Failures Using Bayesian Networks,” *Int. Journal of Precious Engineering Research and Applications*, vol. 1, no. 4, p. 11, 2016.

GINES CARRASCAL was born in Salamanca, Spain in 1975. He received the M.Sc. degree in physics from the University of Salamanca (Spain) in 1999.

From 2000 he has been working as Architect at IBM Consulting Spain, getting involved with quantum computing from 2017, now acting as Quantum Technical Ambassador and Qiskit Advocate. He has become IBM Certified Associate Developer - Quantum Computation using Qiskit v0.2X in 2021.

Since 2014 he has been an Adjunct Professor of computer science at Universidad Carlos III de Madrid and since 2018, Adjunct Professor at Universidad Complutense de Madrid at the computer science departmental section of the Mathematics Faculty. His research interest includes Artificial Intelligence and the application of Quantum Computing to optimization problems, specially but not only in the field of Banking and Financial Services.

Prof. Ginés Carrascal has received two Outstanding Technical Achievement Awards (OTAA), the highest IBM technical award in 2012 and 2016.

GUILLERMO BOTELLA (SM'19) received the M.Sc. degree in Physics (Fundamental) in 1998, the M.Sc. degree in Electronic Engineering in 2001 and the Ph.D. degree (Computer Engineering) in 2007, all from the University of Granada, Spain. He was a research fellow funded by EU working at University of Granada, Spain and the Vision Research Laboratory at University College London, UK. After that, he joined as Assistant Professor at the Department of Computer Architecture and Automation of Complutense University of Madrid, Spain where

he is currently Associate Professor. He has performed research stays acting also as visiting professor from 2008 to 2012 at the Department of Electrical and Computer Engineering, Florida State University, Tallahassee, USA. His current research interests include Image and Video Processing for VLSI, FPGAs, GPGPUs, Embedded Systems, and novel computing paradigms such as analog and quantum computing.

Dr Botella, since 2019 he has become an IEEE Senior Member

ALBERTO A. DEL BARRIO (SM'19) received the Ph.D. degree in Computer Science from the Complutense University of Madrid (UCM), Madrid, Spain, in 2011.

He has performed stays at Northwestern University, University of California at Irvine and University of California at Los Angeles. Since 2021, he is an Associate Professor (tenure-track, civil-servant) of Computer Science with the Department of Computer Architecture and System Engineering, UCM. His main research interests

include Design Automation, Arithmetic, Analog/Quantum Computing and their application to the field of Artificial Intelligence.

Dr Del Barrio is leading the PARNASO project, funded by the Leonardo Grants program by Fundación BBVA. The main objective is to natively integrate the posit format in a hardware/software platform. Since 2019 he is an IEEE Senior Member and since December 2020, he is an ACM Senior Member, too.

DAVID KREMER was born in 1991. He received the Degree in Physics from the Universidad Autónoma de Madrid (Spain) in 2013 and the M.Sc in Mathematical Modelling and Scientific Computing from University of Oxford (United Kingdom) in 2014.

From 2014 he has been working as Data Scientist at IBM Consulting Spain, applying Artificial Intelligence to solve business problems at a wide range of industries, and since 2019 acting as Chief Data Scientist for IBM Consulting Spain. Since 2019 he has been an Adjunct Professor of Artificial

Intelligence at IE University (Madrid) and at Universidad Europea de Madrid. His research interest includes Classical and Quantum Machine Learning and their application to business problems of diverse industries.