



# A Bayesian-network-based quantum procedure for failure risk analysis

Gines Carrascal<sup>1,3\*</sup>, Guillermo Botella<sup>2</sup>, Alberto del Barrio<sup>2</sup> and David Kremer<sup>3</sup>

\*Correspondence:

[ginescar@ucm.es](mailto:ginescar@ucm.es);  
[gines\\_carrascal@es.ibm.com](mailto:gines_carrascal@es.ibm.com)

<sup>1</sup> Department of Informatics  
Systems and Computation,  
Computer Science Faculty,  
Complutense University of Madrid,  
Madrid, 28040, Spain

<sup>3</sup> IBM Consulting España, Madrid,  
28830, Spain

Full list of author information is  
available at the end of the article

## Abstract

Studying the propagation of failure probabilities in interconnected systems such as electrical distribution networks is traditionally performed by means of Monte Carlo simulations. In this paper, we propose a procedure for creating a model of the system on a quantum computer using a restricted representation of Bayesian networks. We present examples of this implementation on sample models using Qiskit and test them using both quantum simulators and IBM Quantum hardware. The 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.

**Keywords:** Bayesian network; Quantum computing; Risk analysis; Resilience analysis; Reliability analysis

## 1 Introduction

Efficient electricity distribution networks constitute a crucial component of modern society today. Therefore, the grid must not only be reliable, but also resilient. Consequently, modelling such systems for the purpose of mitigating possible failures and power outages is an important area of research [1].

Resilience planning starts with a 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. In this study, we consider electrical substations as one of the most important components of an electric power distribution network, and we know that 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 [2]. Using aging models in combination with data from sensors on the elements themselves, we can calculate the probability of a given element failing in the distribution network. In this study we 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, and 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 [3]. The difficulty associated with these

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

models is their high computational complexity. As the size of the problem increases, and 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 required by classical algorithms to solve these Bayesian network models grow exponentially, up to a point at which their resolution becomes classically unfeasible [4].

In this regard, it is worth noting that gate-based quantum computers are expected to help solve problems in quantum chemistry [5–7], machine learning [8, 9], financial simulation [10–13] and the application of combinatorial optimization [14, 15].

As predicted by Preskill [16], 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 limits 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.

Quantum advantage refers to the ability of a quantum computer to solve a problem faster or more efficiently than a classical computer [17, 18]. Achieving quantum advantage is a major goal and challenge for quantum computing, as it would demonstrate the practical benefits and applications of this emerging technology.

At this point we can highlight the main contributions in this work:

- We define a new Restricted Quantum Bayesian Network (RQBN) procedure for modelling the reliability of complex systems.
- We evaluate the viability of the procedure for performing reliability analysis, obtaining the same accuracy as classical Monte Carlo methods by adjusting the number of shots in a single quantum circuit execution.
- We test the impact of real quantum computer noise on the elements in the model.

This paper proposes a procedure for the general application of the quantum modelling of the propagation of failure probabilities in 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 substation and a protection system) are modelled by Bayesian networks as examples. The failure modes of the substations are calculated with both classical algorithms and quantum algorithms.

We approach the solving of Bayesian networks in a classical way using *pomegranate*, a Python library that is able to implement probabilistic models [19]. We then look to model and solve Bayesian networks in a quantum regime using Qiskit [20], in view of its lower code effort [21].

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

We conclude by discussing the future directions and open questions for achieving quantum advantage for failure risk analysis. Currently, quantum computing for this problem is not completely solved. There are still gaps in scaling and error mitigation. To bridge these gaps, we propose or suggest exploring other methods to build Quantum nodes (i.e., quantum neurons [22, 23]) that could enhance our quantum solution or enable new quantum solutions. However, there are also limitations or barriers that prevent or hinder achieving

quantum advantage for this problem, such as the length of the circuits and the coherence time of the current quantum computers. Therefore, we identify and highlight some open questions or challenges that need to be addressed or solved for achieving quantum advantage for this problem.

The rest of the paper is organized as follows. Section 2 reviews the foundations needed by the Restricted Quantum Bayesian Network model described in Sect. 3.1. Sections 3.2 and 3.3 explore two case studies for the defined procedure. Section 3.4 discusses the implications of noise in real quantum computers, and Sect. 4 contains the conclusions.

## 2 Methods

### 2.1 Reliability analysis

The resilience of a system is related to its ability to withstand stress, adapt to this stress, avoid disruptions, and recover from them. Resilience is linked to the traditional concept of reliability but encompasses additional concepts. Planning for resilience requires taking into account both the risks and the costs of mitigating those risks [1].

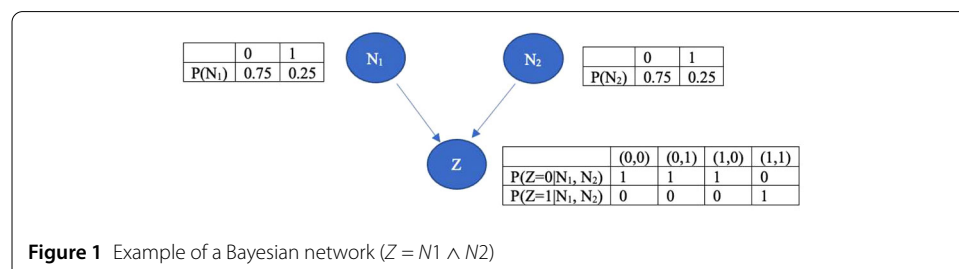
Electrical substation reliability has been approached classically by many authors using different methods, including fault tree and event tree analysis [24], failure modes and effect evaluation [25], and also Bayesian networks [4], and with this the reliability of the system can be obtained using probability propagation techniques. This allows the modelling of complex systems, such as those of the bridge type, and dependencies between failures, which are difficult to obtain with conventional reliability analysis techniques.

Failure risk analysis is a problem that involves high-dimensional probabilistic inference on complex systems, such as electrical distribution networks. Classical methods, such as Monte Carlo simulations, can be computationally expensive and inaccurate for such problems, especially when dealing with large-scale networks with many variables and dependencies. Quantum computers, on the other hand, can exploit the principles of quantum mechanics, such as superposition and entanglement, to manipulate information in ways that are not possible for classical computers. This could potentially give them an edge for solving problems that are very complex, such as optimizing large-scale networks.

### 2.2 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 make it up.

In our problem we only work with discrete variables so we only model discrete Bayesian networks. The strength of the relationship between the variables is quantified by conditional probability tables (CPTs) associated with each node.



The only restriction on the arcs connecting the 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 and is much like representing a flow of information through the network.

To perform 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 use a Python library called *pomegranate* [26], which 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 are usually applied, but, in general, both exact and approximate algorithms theoretically have exponential complexity and so the problem is still considered to be NP hard [27].

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

### 2.3 Quantum gates

To be able to model these conditional probabilities, we 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:

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

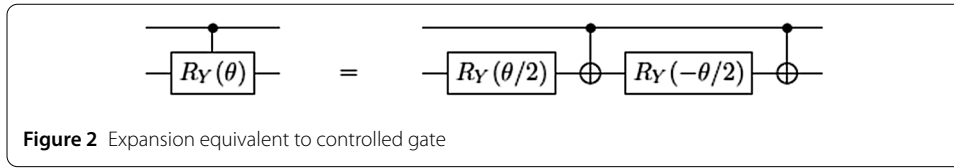
***R<sub>Y</sub> gate*** This gate corresponds to a rotation of  $\theta$  radians around the y-axis of the Bloch sphere. It is represented as:

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

***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 to the target qubit to change its state. On the computational basis with states  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$ , and  $|11\rangle$ , the matrix representing the CNOT gate is as follows:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3)$$



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  $|1\rangle$ . It is represented as:

$$CR_Y(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ 0 & 0 & \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}. \quad (4)$$

*Three qubit and other quantum gates* The two three-qubit gates we use for our algorithm are the CCNOT (CCX or Toffoli) and the  $CCR_Y(\theta)$  gates. 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  $R_Y(\theta)$  gate 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 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 (Fig. 2).

### 3 Results and discussion

#### 3.1 Restricted quantum Bayesian network model

In a general way, Sima E. Borujeni et al. established a method for mapping conditional probabilities from a Bayesian network to a quantum circuit [28], 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 failure probability of the electrical grid, we have a particular case of the Bayesian network in which the leaf nodes are the random variables representing the failure probability of each individual element. The rest of the nodes can be modeled with combinations of “AND” and “OR” nodes, depending on the serial or parallel configuration of the elements, respectively.

Under these circumstances, in this paper, we define what we call a Restricted Quantum Bayesian Network (RQBN) that leads to a lower depth circuit than the generic implementation, with only one operation per node. This formation is more suitable for being run on current NISQ-type quantum hardware.

To solve this problem in the quantum regime, we employ the following 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 refer to 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.

**Root nodes** To represent a root node with two states we apply an  $RY(\theta)$  gate with an appropriate angle  $\theta$ . Thus, the probabilities of the root nodes can be assigned to the probabilities (and therefore to the probability amplitudes) of the basic states  $|0\rangle$  and  $|1\rangle$ . Let  $\theta_{Vi}$  be the angle representing the angle of rotation associated with a two-state node  $V_i$ . If the initial state of a qubit is  $|0\rangle$ ,  $RY(\theta)$  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:

$$\theta_{V_i} = 2 \times \tan^{-1} \sqrt{\frac{P(|1\rangle)}{P(|0\rangle)}} = 2 \times \tan^{-1} \sqrt{\frac{P(V_i = 1)}{P(V_i = 0)}}. \quad (5)$$

**“AND” node for serial elements** A serial system like the one in Fig. 3 can be modeled with a Bayesian network such as the one in Fig. 1.

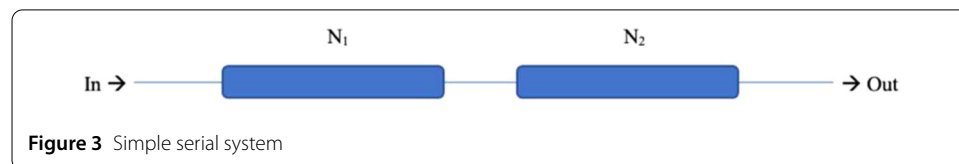
Nodes  $N_1$  and  $N_2$  refer to elements 1 and 2 of the system in series, while node  $Z$  corresponds to the probability that the entire system will continue to operate. Each element that is part of the system has a probability of operation of  $P(N_i) = p_i$  and a failure probability of  $P(\neg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

These probabilities are estimated by domain experts and are calculated by considering several factors such as the type of element involved, the length of time it has been running, or whether it has undergone 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 conditional probability table shown in Table 1.

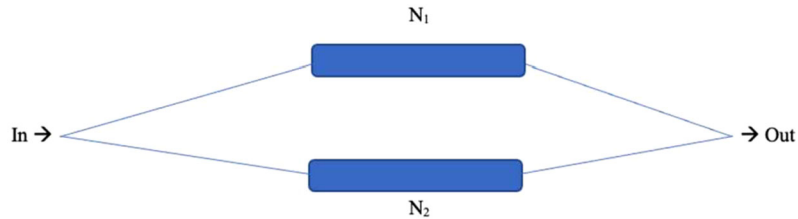
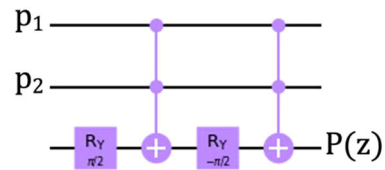
In a serial system, we need the operation of both elements for the system to be successful, so we can use the Conditional Probability Table (CPT) of node  $Z$  as a truth table of a logical conjunction. Thus, the reliability of this system (i.e., the probability that the system will work) is:

$$P(Z) = 1 \cdot p_1 p_2 + 0 \cdot q_1 p_2 + 0 \cdot p_1 q_2 + 0 \cdot q_1 q_2. \quad (6)$$



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

**Figure 4** “AND” node for 2 control qubits**Figure 5** Simple parallel system**Table 2** 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

And the probability of system failure is:

$$P(z) = 0 \cdot p_1 p_2 + 1 \cdot q_1 p_2 + 1 \cdot p_1 q_2 + 1 \cdot q_1 q_2. \quad (7)$$

We can implement this behavior as a multiple controlled rotation of  $\pi$  radians. The multiple controlled gates provide this  $P(z)$  desired behavior, depending on the values of the probabilities of the control qubits ( $p_1, p_2$ ).

We use the expansion of controlled gates to obtain the circuit in Fig. 4.

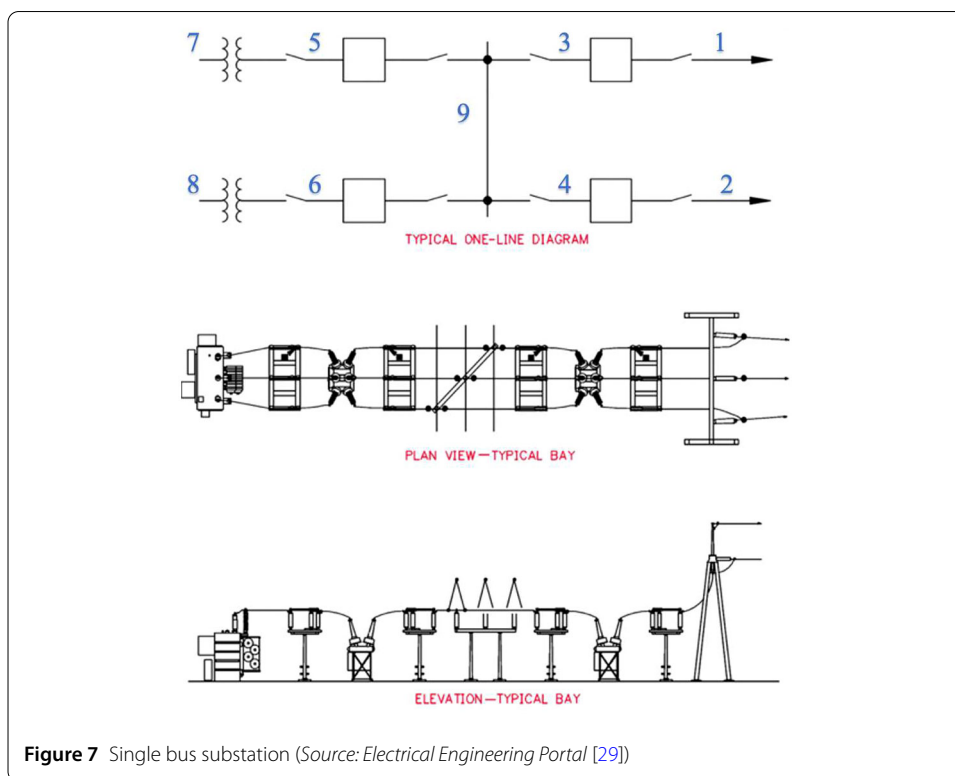
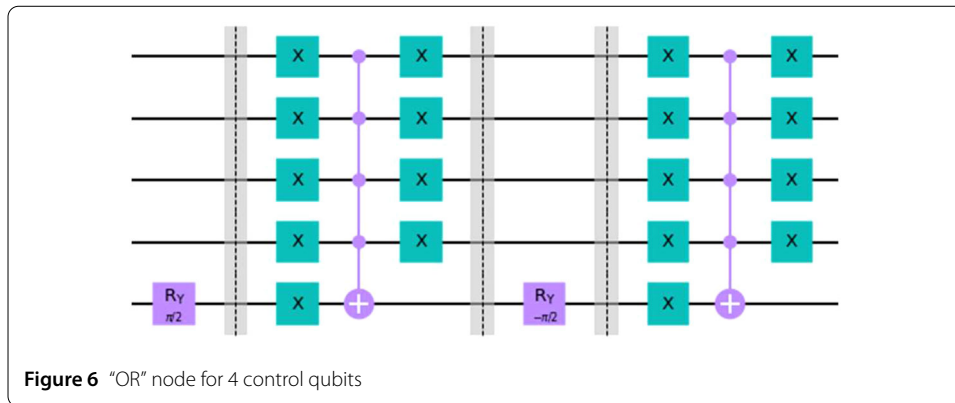
**“OR” node for parallel elements** Similarly, we can explain how to implement a parallel system such as the one shown in Fig. 5.

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 2.

We can implement this behavior as an expansion in a similar way to that previously used for the “AND” node, but in this case substituting the Toffoli gates with a circuit implementing the OR logic, as depicted in Fig. 6.

The circuit in Fig. 6 uses 5 qubits: 4 for the nodes representing the inputs and one for the node representing the output (remember, a quantum circuit must be reversible). In this circuit, the column of  $X$  gates before the multiple CNOT ensures any input different from ‘0000’ will result in the flip of the target qubit, and the second column of  $X$  will return the input qubits to their original state.

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



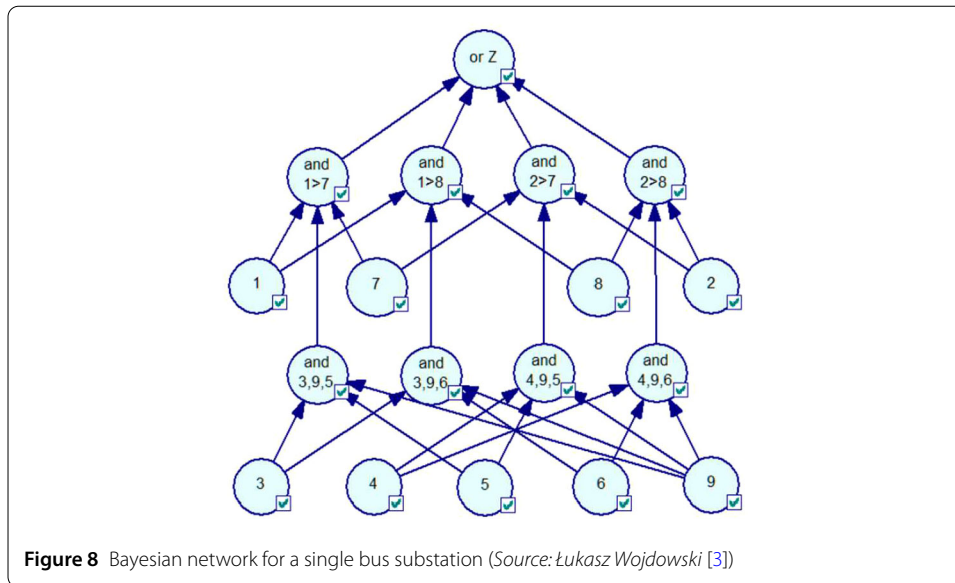
*Finding most common failure modes* By using this representation, we can obtain the probability of each combination of failure scenarios for the root elements, as well as the corresponding state of the combination nodes, including the final node that represents the failure of the complete circuit. Thus, it is straightforward to obtain all the measurements that represents a failure looking at the value of the final node, and the cause of the failure, simply by observing the results corresponding to each of the other individual elements.

### 3.2 Case study: substation failure

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

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





We can perform a path analysis to create a Bayesian network representing this substation configuration [3] as follows.

Each individual element in the substation can 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, the environmental conditions, and sensor measurements, amongst other factors.

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

To model the complete system status, all the paths must be combined at an “OR” node, representing the case in which all these paths are possible parallel circuits for the electricity flow. The resulting Bayesian network is shown in Fig. 8.

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

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

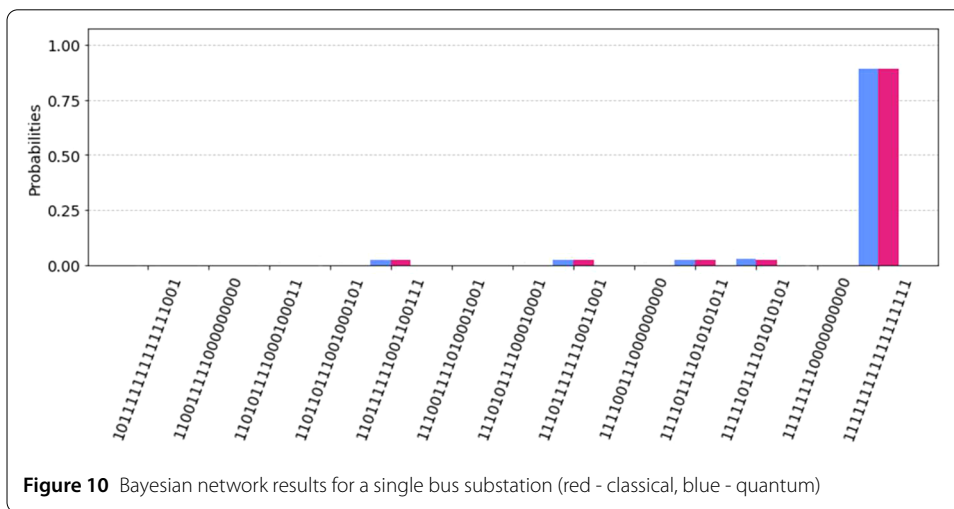
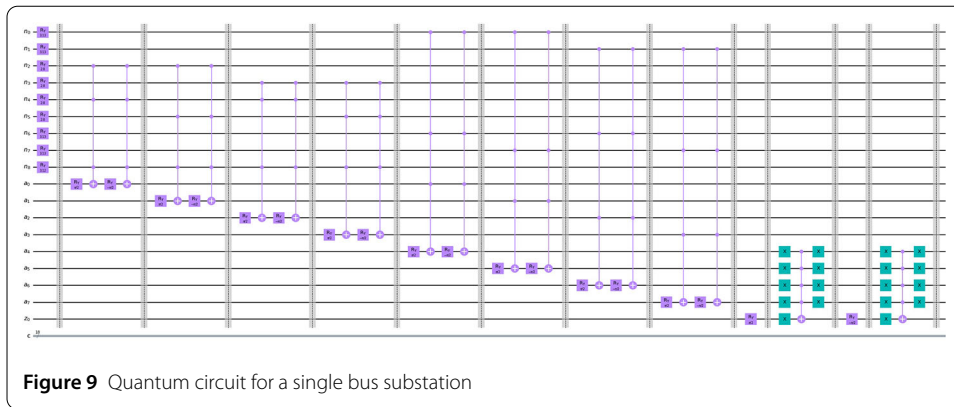
Each state of the system is represented by a list of ones and zeroes, each one corresponding to the state (working = 1 or failing = 0) of each node in the Bayesian network. The first elements correspond to the individual elements in the circuit, and then the following ones are the dependent nodes.

The last one corresponds to the last node in the network, which represents whether the system is working or not.

To obtain the classical probabilities, we used each of the possible states as an input and ran a loop on the *pomegranate* model to calculate the probability of each one.

To obtain the quantum probabilities, we ran the quantum circuit once and measured the probabilities of each possible state.

The resulting circuit is shown in Fig. 9. The module in charge of the measurements in the circuit (one measurement for each qubit) is not shown to reduce the size of the image.



A comparison of the probabilities of each mode obtained classically and with the quantum model is shown in Fig. 10.

The numbers of the percentages in the bars are not essential in this figure as the idea is just to see that the shapes are similar.

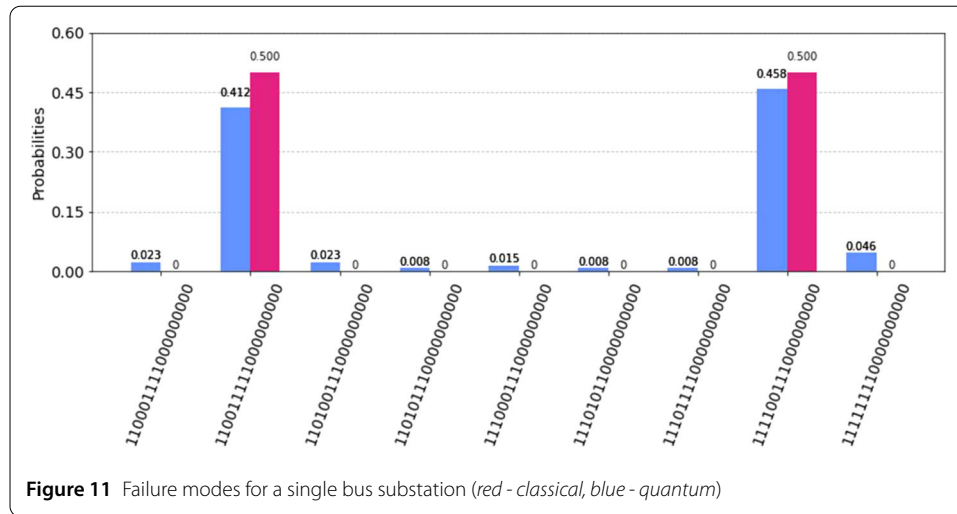
In Fig. 10 the results ending in “1” correspond to “working modes”, and the results ending in “0” correspond to “failure modes”.

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

### 3.3 Case study: protection systems

**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 of a set of “protections”, which are digital or analogue units that perform different monitoring functions. Typically, each protection unit can per-



form 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 case study is to estimate the failure probabilities 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 failure probabilities of the individual units in the system. Specifically, we searched for the most probable combination of unit failures that can lead to a system failure, and for the combination that involves the minimum number of failing units that then leads to a system failure. These aforementioned functions have been categorized as “essential” and “non-essential”. 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.

Or:

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.

We could not find publicly available data sets, that contain both the failure probabilities and the dependencies of the components in an electrical distribution network. Moreover, we had to deal with the issues of data quality, privacy, and security when working with real-world data. Therefore, we decided to use synthetic data for our experiments, which allowed us to control the parameters and test our procedure under different scenarios.

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. However, we acknowledge that using synthetic data limits the generalizability and applicability of our results, emphasizing the need for further experiments with real-world data in the future.

*Solution approach 1: exact calculation* A straightforward way to compute the failure probability of the system is to enumerate all combinations of unit failures, compute the probability of each of these and then determine, for each unit, whether they result in a system failure based on the failure rules described previously.

For this purpose, 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 whether 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 failure probability 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 on modern hardware (i.e., public cloud resource intensive working nodes, suitable for HPC [30]), both in terms of memory and CPU time. It is worth noting that systems with more than 20 units are not uncommon in practice, and they can even have more than 50.

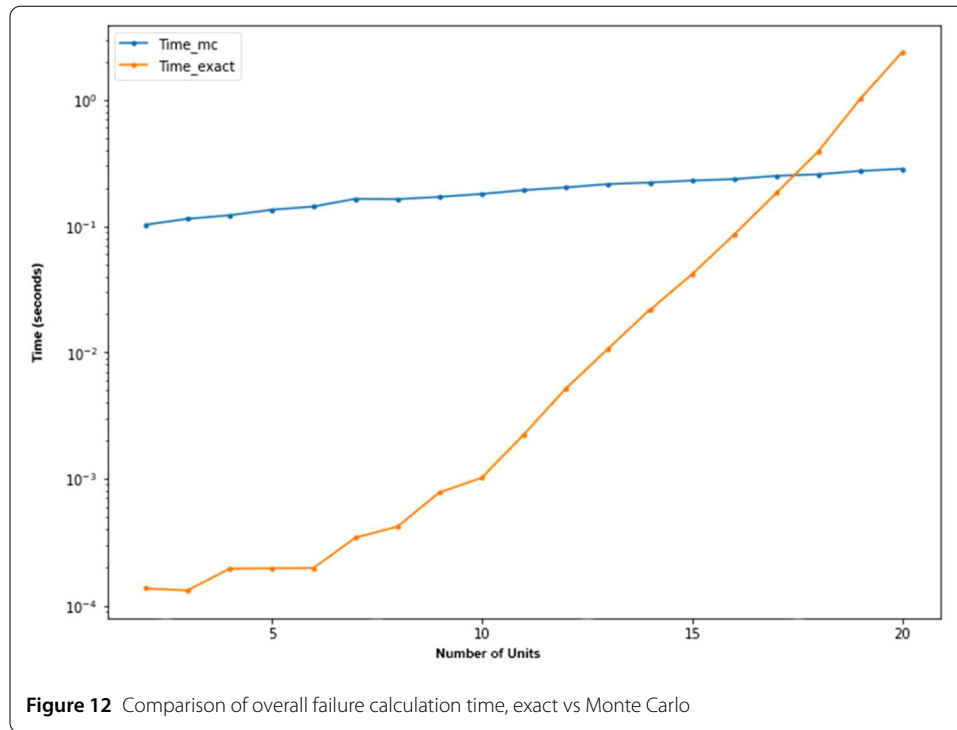
*Solution approach 2: Monte Carlo estimation* In this Monte Carlo approach, instead of considering all possible samples, random samples are generated according to their probability of occurrence. As in the exact method, we then determine whether each of the samples generated results in a system failure 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 Fig. 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).

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

*Solution approach 3: quantum Bayesian networks* We can model the behavior of the protection systems with a Restricted Quantum Bayesian Network. The root nodes can represent the individual units with an  $R_Y$  gate rotating the qubit in order to represent the failure probability of the corresponding unit.

A first layer of AND nodes aggregates each of the protection functions, and on a second layer, one AND node aggregates the non-essential functions, and an OR node the essen-



tial ones. Each root node is connected only to those nodes representing the functions protected by this unit.

On the top layer, a final AND node aggregates the previous two nodes, representing the overall status of the system.

Figure 13 shows an example of a Bayesian network for 10 protection units providing 4 non-essential functions and 7 essential ones.

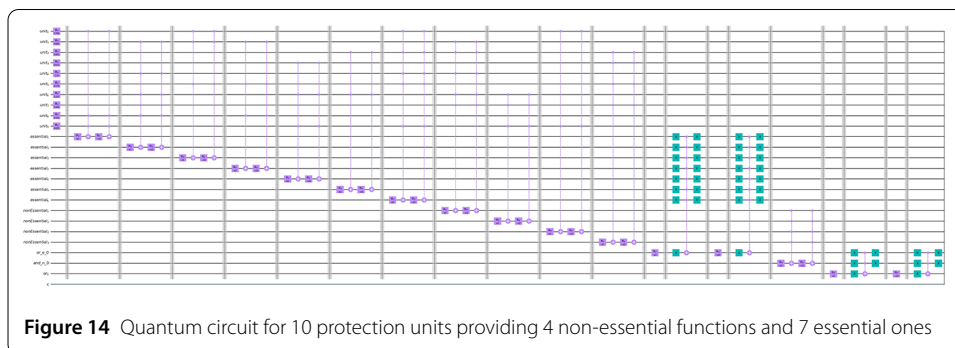
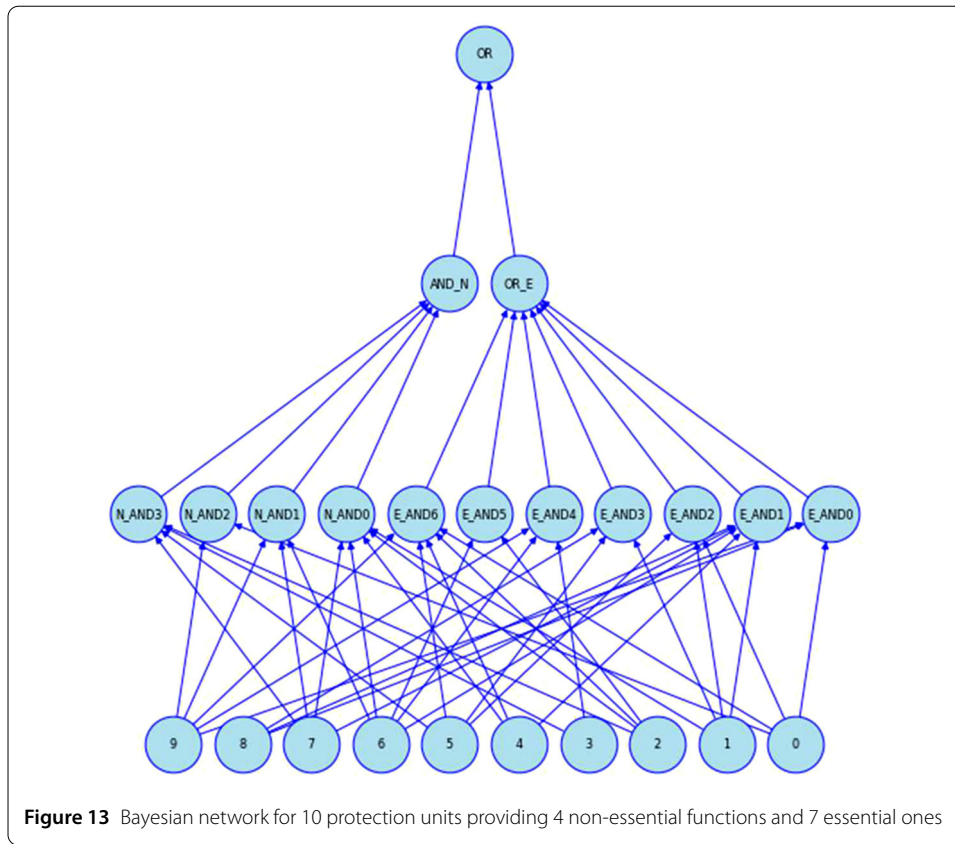
The lower row of nodes (root nodes) represent the failure probability of each protection system. In the second row, the nodes starting with E\_ represent the essential functions and those starting with N\_ the non-essential ones. In the third row the logic of all the non-essential functions failing (AND\_N), or any of the essential functions failing (OR\_E) is implemented. The final “OR” node represents the overall behavior of the system.

Figure 14 provides a representation of the quantum circuit developed from the network in Fig. 13. Again, to reduce the size of the image, the measurement modules have been omitted.

In order to provide a comparison with the Monte Carlo approach, a random system with 16 protection units providing 7 essential functions and 4 non-essential functions was also generated, and 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 Monte Carlo and with the Restricted Quantum Bayesian Network (RQBN) approaches. Figure 15 shows the mean error compared with the exact calculation for both the Monte Carlo and the RQBN approach.

It is important to note that in Fig. 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. Thus, we can adjust the desired accuracy of our quantum method with the

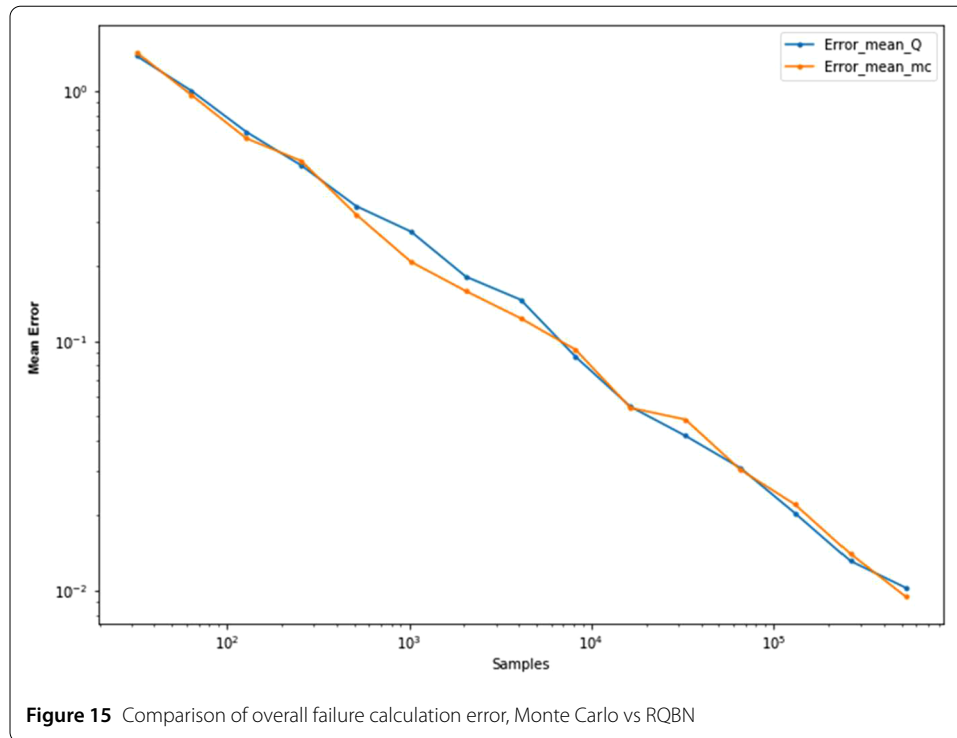


number of shots to match the corresponding accuracy of the Monte Carlo method with a given number of iterations.

### 3.4 Execution on real quantum computers

*Limited number of qubits* To validate the feasibility of the RQBN procedure on real NISQ quantum computers, and to simplify the comparison of the experimental results, we evaluated the Bayesian network shown in Fig. 8 on several Quantum computers provided by IBM's Quantum cloud services.<sup>1</sup>

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



*Noise implications* On NISQ computers, noise accumulates with the depth of the circuits and affects the capability of theoretical algorithms to speed up the process of solving realistic problems.

Noise imposes a natural constraint on accessible circuit depths, and it scales exponentially to penalize greater 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 should not forget that equivalent quantum circuits for the Bayesian networks may already result in better quality quantum computers than the ones available today. Observing these quantum speedups in practice is not yet possible.

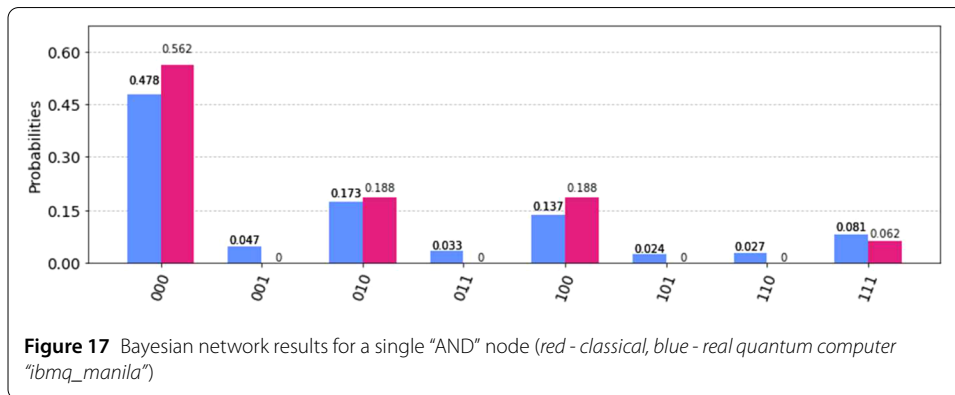
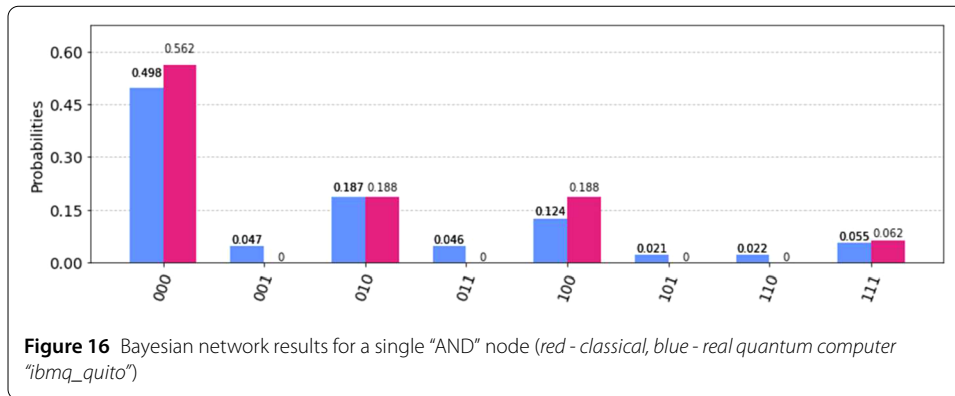
*Experimental results* We tested a simple Bayesian network consisting of only one “AND” node on two different real quantum computers: “ibmq\_quito”, with 5 qubits and Quantum Volume 16 (CNOT average error  $1.337\text{e-}2$ ), and “ibmq\_manila”, also with 5 qubits but Quantum Volume 32 (CNOT average error  $7.673\text{e-}3$ ).

In both cases the measurements were taken with 8182 shots. The results have shown in Fig. 16 and Fig. 17. For purposes of comparison, we have added the red bars, which represent the theoretical exact results. As one can see, due to errors, some results that should be 0 appear with a non-zero probability in the results from the real quantum computers (blue bars).

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

To test the performance of our quantum algorithm on different hardware platforms, we also ran our experiments on another quantum processor, “ibmq\_hanoi”. This is a 27-qubit





Falcon processor with a quantum volume of 64 at the time of this experiment. However, we were not able to obtain systematically significant results with this processor, as the circuits were too long for the current coherence time. This indicates that our quantum algorithm requires further optimization and error mitigation techniques to run successfully on noisy intermediate-scale quantum (NISQ) devices.

IBM has recently announced its goal of building a tool capable of calculating unbiased observables of circuits with 100 qubits and depth-100 gate operations in a reasonable run-time by the end of 2023 [31], which would be a major milestone for the quantum industry. Achieving this goal would require overcoming many technical challenges, such as improving qubit quality, connectivity, and coherence, as well as developing efficient algorithms and error correction schemes. Our quantum algorithm is one example of how we can leverage the power of quantum computing to solve complex optimization problems, but it also shows the limitations and difficulties of running such algorithms on current NISQ devices.

We are also working on entanglement forging [32] and circuit knitting [33] to reduce the length of the circuits. Entanglement forging is a technique that uses a classical computer to capture quantum correlations and effectively split the problem in half, making it possible to run a given quantum circuit using only half as many qubits on a quantum computer. This reduces the quantum resource overhead and improves the accuracy. However, entanglement forging requires some conditions to be met, such as weak entanglement between the two halves of the original system.

Circuit knitting is a process of decomposing a quantum circuit into smaller circuits, executing those smaller circuits on quantum processors, and then knitting their results into a



reconstruction of the original circuit's outcome. Circuit knitting includes techniques such as circuit cutting, and classical embedding. Circuit knitting can help reduce the circuit length by partitioning large quantum circuits into subcircuits that fit on smaller devices, at the cost of a calculation overhead.

Also, as future lines of research, we could use variational quantum circuits or quantum machine learning techniques (i.e., quantum neurons [22, 23]) to construct more expressive and adaptive quantum nodes that could capture more complex dependencies and functions. This could potentially improve the accuracy and efficiency of our procedure for failure risk analysis.

However, we also note that using other methods of building quantum nodes could introduce new challenges and trade-offs. For example, we would need to find optimal ways to train and optimize the quantum nodes, to deal with the noise and errors in the quantum hardware, and to balance the trade-off between expressivity and complexity of the quantum nodes.

## 4 Conclusions

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

RQBN is a novel and efficient way to encode the system variables into quantum states, to implement the Bayesian network structure and operations on a quantum circuit, and to extract meaningful information from the quantum measurements. We show that our procedure can scale well with the system size. We also show that our procedure can be implemented on both quantum simulators and IBM Quantum hardware using Qiskit, a quantum software framework.

We have implemented this model for different scenarios and validated the viability of such a 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 as the number of iterations of the classical Monte Carlo simulation. This provides some preliminary results that suggest that our procedure could achieve quantum advantage for failure risk analysis in the future. Mainly because quantum computers are still in the early stages of developing and, even though there are some of them with more than 100 qubits, 1) their access is very restricted even for IBM employees and, 2) they still suffer from noise, so in practice it is not possible yet to perform large-scale experiments with coherent results.

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

### Funding

This work has been supported by grant PID2021-123041OB-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe", and by the CM under grant S2018/TCS-4423.

### Abbreviations

CPT, Conditional Probability Table; DAG, Directed Acyclic Graph; NISQ, Noisy Intermediate-Scale Quantum; RQBN, Restricted Quantum Bayesian Network.

### Availability of data and materials

All data generated or analysed during this study are included in this published article.

## Declarations

### Ethics approval and consent to participate

Not applicable. Research not involving human subjects

### Consent for publication

Not applicable. Research not involving human subjects

### Competing interests

The authors declare no competing interests.

### Author contributions

GC, GB, AB, and DK designed all the experiments. The manuscript was written with contributions from all authors. All authors read and approved the final manuscript.

### Authors' information

**Gines Carrascal** was born in Salamanca, Spain in 1975. He received an M.Sc. degree in Physics from the University of Salamanca (Spain) in 1999. Since 2000 he has been working as Architect at IBM Consulting Spain, becoming involved with quantum computing in 2017, now acting as Quantum Technical Ambassador and Qiskit Advocate. He is an 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, especially but not only in the field of Banking and Financial Services. **Guillermo Botella** received an M.Sc. degree in Physics (Fundamental) in 1998, an M.Sc. degree in Electronic Engineering in 2001 and a Ph.D. degree (Computer Engineering) in 2007, all from the University of Granada, Spain. He was a research fellow funded by the EU working at the University of Granada, Spain and the Vision Research Laboratory at University College London, UK. After that, he joined the Department of Computer Architecture and Automation as Assistant Professor at the Complutense University of Madrid, Spain, where he is currently Associate Professor. He has had research stays, also acting 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. **Alberto del Barrio** received a Ph.D. degree in Computer Science from the Complutense University of Madrid (UCM), Madrid, Spain, in 2011. He has had stays at Northwestern University, University of California at Irvine, and University of California at Los Angeles. Since 2021, he has been 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 has been the PI of the PARNASO project, funded by the Leonardo Grants program by Fundación BBVA, and currently he is the PI of the ASIMOV project, funded by the Spanish MCIN, which includes a work package to research the applicability of Quantum Computing. **David Kremer** was born in 1991. He received a Degree in Physics from the Universidad Autónoma de Madrid (Spain) in 2013 and an M.Sc. in Mathematical Modelling and Scientific Computing from the University of Oxford (United Kingdom) in 2014. From 2014 he has been working as Data Scientist at IBM Consulting Spain, applying Artificial Intelligence to solving business problems in a wide range of industries, and since 2019 he has been 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 in diverse industries.

### Author details

<sup>1</sup>Department of Informatics Systems and Computation, Computer Science Faculty, Complutense University of Madrid, Madrid, 28040, Spain. <sup>2</sup>Department of Computer Architecture and Automation, Computer Science Faculty, Complutense University of Madrid, Madrid, 28040, Spain. <sup>3</sup>IBM Consulting España, Madrid, 28830, Spain.

Received: 26 August 2022 Accepted: 24 April 2023 Published online: 08 May 2023

## References

1. Interconnection, P. J. M. PJM's evolving resource mix and system reliability. March 30 (2017): 2017. [https://legacy-assets.eenews.net/open\\_files/assets/2017/03/31/document\\_ew\\_02.pdf](https://legacy-assets.eenews.net/open_files/assets/2017/03/31/document_ew_02.pdf). Accessed 16th, April 2023.
2. MANINT: a digital revolution serving the transmission grid | Red Eléctrica. 2023. <https://www.ree.es/en/pressoffice/monographs/2019/06/manint-digital-revolution-serving-transmission-grid>. Accessed 16th, April 2023.
3. Wojdowski Ł. Substation reliability evaluation with dependent outages and switching failures using Bayesian networks. *Int J Precious Engineering Research and Applications*. 2016;1(4):11.
4. Torres-Toledano JG, Sucar LE. Bayesian networks for reliability analysis of complex systems. In: *Progress in artificial intelligence—IBERAMIA 98: 6th ibero-American conference on AI Lisbon, proceedings 6*. Portugal. Berlin, Heidelberg: Springer; 1998. p. 5–9.
5. Moll N, Barkoutsos P, Bishop LS, Chow JM, Cross A, Egger DJ, Filipp S, Fuhrer A, Gambetta JM, Ganzhorn M, Kandala A. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci Technol*. 2018;3(3):030503. <https://doi.org/10.1088/2058-9565/aab822>.
6. Kandala A, Temme K, Córcoles AD, Mezzacapo A, Chow JM, Gambetta JM. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*. 2019;567(7749):491–5. <https://doi.org/10.1038/s41586-019-1040-7>.
7. Ganzhorn M, Egger DJ, Barkoutsos P, Ollitrault P, Salis G, Moll N, Roth M et al. Gate-efficient simulation of molecular eigenstates on a quantum computer. *Phys Rev Appl*. 2019;11(4):044092. <https://doi.org/10.1103/PhysRevApplied.11.044092>.

8. Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. *Nature*. 2017;549(7671):195–202. <https://doi.org/10.1038/nature23474>.
9. Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, Gambetta JM. Supervised learning with quantum-enhanced feature spaces. *Nature*. 2019;567(7747):209–12.
10. Egger DJ, Gambella C, Marecek J, McFaddin S, Mevissen M, Raymond R, Simonetto A, Woerner S, Yndurain E. Quantum computing for finance: state-of-the-art and future prospects. *IEEE Trans Quantum Eng*. 2020;1:1–24. <https://doi.org/10.1109/TQE.2020.3030314>.
11. Woerner S, Egger DJ. Quantum risk analysis. *npj Quantum Inf*. 2019;5(1):15. <https://doi.org/10.1038/s41534-019-0130-6>.
12. Rebentrost P, Gupta B, Bromley TR. Quantum computational finance: Monte Carlo pricing of financial derivatives. *Phys Rev A*. 2018;98(2):022321. <https://doi.org/10.1103/PhysRevA.98.022321>.
13. Egger DJ, Gutiérrez RG, Mestre JC, Woerner S. Credit risk analysis using quantum computers. *IEEE Trans Comput*. 2020;70(12):2136–45. <https://doi.org/10.1109/TC.2020.3038063>.
14. Braine L, Egger DJ, Glick J, Woerner S. Quantum algorithms for mixed binary optimization applied to transaction settlement. *IEEE Trans Quantum Eng*. 2021;2:1–8. <https://doi.org/10.1109/TQE.2021.3063635>.
15. Barkoutsos PK, Nannicini G, Robert A, Tavernelli I, Woerner S. Improving variational quantum optimization using CVaR. *Quantum*. 2020;4:256. <https://doi.org/10.22331/q-2020-04-20-256>.
16. Preskill J. Quantum computing in the NISQ era and beyond. *Quantum*. 2018;2:79. <https://doi.org/10.22331/q-2018-08-06-79>.
17. Quantum Advantage, Supremacy and What You Need to Know (2021). <https://www.quantumcomputinginc.com/blog/quantum-advantage/>. Accessed 10th Apr 2023.
18. Daley AJ, Bloch I, Kokail C, Flannigan S, Pearson N, Troyer M, Zoller P. Practical quantum advantage in quantum simulation. *Nature*. 2022;607(7920):667–76. <https://doi.org/10.1038/s41586-022-04940-6>.
19. Pomegranate 0.13.2 documentation. <https://pomegranate.readthedocs.io/en/latest/>. Accessed 16th April 2023.
20. Anis, Sajid M, Abraham H, Agarwal AduOffei R, Agliardi G, Aharoni M, Akhalwaya IY, Aleksandrowicz G, Alexander T, Amy M, Anagolum S. Qiskit: An open-source framework for quantum computing. Qiskit/qiskit. 2021.
21. Carrascal G, Del Barrio AA, Botella G. First experiences of teaching quantum computing. *J Supercomput*. 2021;77:2770–99. <https://doi.org/10.1007/s11227-020-03376-x>.
22. Mangini S, Tacchino F, Gerace D, Macchiavello C, Bajoni D. Quantum computing model of an artificial neuron with continuously valued input data. *Mach Learn: Sci Technol*. 2020;1(4):045008. <https://doi.org/10.1088/2632-2153/abaf98>.
23. Torrontegui E, García-Ripoll JJ. Unitary quantum perceptron as efficient universal approximator (a). *Europhys Lett*. 2019;125(3):30004. <https://doi.org/10.1209/0295-5075/125/30004>.
24. Wang F, Tuinema BW, Gibescu M, van der Meijden MA. Reliability evaluation of substations subject to protection system failures. In: 2013 IEEE Grenoble conference. 2013, June. p. 1–6. IEEE.
25. Nack D. Reliability of Substation Configurations. 2005 (Iowa State University) <http://www.ee.umn.edu/class/ee5725/SubstationReliability.pdf>. Accessed 16th April 2023.
26. Schreiber J. Pomegranate: fast and flexible probabilistic modeling in python. *J Mach Learn Res*. 2017;18(1):5992–7.
27. Chickering M, Heckerman D, Meek C. Large-sample learning of Bayesian networks is NP-hard. *J Mach Learn Res*. 2004;5:1287–330.
28. Borujeni SE, Nannapaneni S, Nguyen NH, Behrman EC, Steck JE. Quantum circuit representation of Bayesian networks. *Expert Syst Appl*. 2021;176:114768. <https://doi.org/10.1016/j.eswa.2021.114768>.
29. Csanyi E. 6 common bus configurations in substations up to 345 kV | EEP. In: EEP - electrical engineering portal. 2019. <https://electrical-engineering-portal.com/bus-configurations-substations-345-kv>. Accessed 16th April 2023.
30. Compute optimized instances - Amazon Elastic Compute Cloud. 2023. <https://docs.aws.amazon.com/ec2/index.html>. Accessed 16th April 2023.
31. Quantum-centric supercomputing: The next wave of computing. IBM Research Blog. 2021. <https://research.ibm.com/blog/next-wave-quantum-centric-supercomputing>. Accessed 16 Apr 2023.
32. Eddins A, Motta M, Gujarati TP, Bravyi S, Mezzacapo A, Hadfield C, Sheldon S. Doubling the size of quantum simulators by entanglement forging. *PRX Quantum*. 2022;3(1):010309. <https://doi.org/10.1103/PRXQuantum.3.010309>.
33. Piveteau C, Sutter D. Circuit knitting with classical communication. 2022. Preprint. [arXiv:2205.00016](https://arxiv.org/abs/2205.00016).

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.