mathematics

MDPI

Article

Estimating the Number of Junta Variables for Optimizing Boolean Functions in Quantum Memories

Abdulaziz Alotaibi [1], Samar AbdelAzim [2], Sattam Saleem Alharbi [1] and Mohamed Darwish [3,*]

[1] Department of Mathematics, College of Science and Humanities in AlKharj, Prince Sattam Bin Abdulaziz University, AlKharj 11942, Saudi Arabia; am.alotaibi@psau.edu.sa (A.A.); sat.alharbi@psau.edu.sa (S.S.A.)
[2] Department of Computer Science, Faculty of Computers and Information, Assuit University, Assuit 71515, Egypt; samar1992@compit.aun.edu.eg
[3] Department of Computer Science, University College of Al Wajh, University of Tabuk, Tabuk 71491, Saudi Arabia
* Correspondence: m_darwish@ut.edu.sa

Abstract

Optimizing Boolean function components to have the minimum number of inputs in order to reduce the memory space required during these functions in computing devices is a significant demand. This paper proposes a quantum computation approach based on the degree-of-entanglement quantum computation model to estimate the number of junta variables of an unknown Boolean function presented through an oracle. The time complexity of the developed quantum approach is independent of the number of inputs and depends on an allowable assigned error ϵ. Thus, the time complexity of the developed algorithm is $O(\epsilon^{-2})$, compared to $O(2^{n+1})$ in the traditional approach. Also, the memory space of the developed approach is linear, $O(2n+4)$, in terms of the number of inputs compared to the exponential memory space $O(2^{n+1})$ using the traditional approach. Therefore, the developed quantum approach has exponential supremacy in comparison to the traditional approach. The developed approach was implemented practically using both the Qiskit simulator and the IBM real quantum computer. The obtained results expose high statistical fidelities between the empirical and theoretical results.

Keywords: quantum random access memory (qRAM); data structure; quantum information; quantum computing

MSC: 81P68

check for updates

Academic Editor: Raymond Lee

Received: 6 September 2025
Revised: 19 October 2025
Accepted: 20 October 2025
Published: 25 October 2025

Citation: Alotaibi, A.; AbdelAzim, S.; Alharbi, S.S.; Darwish, M. Estimating the Number of Junta Variables for Optimizing Boolean Functions in Quantum Memories. *Mathematics* **2025**, *13*, 3400. https://doi.org/10.3390/math13213400

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recently, the most efficient quantum main memory was developed [1,2], which can access all the data stored in its memory cells with a time cost of $O(1)$. Additionally, it can store chunks of data in all memory cells with a time cost of $O(1)$. Moreover, this memory reduces the memory space that is required to store an amount of data exponentially compared to classical RAM [1,2]. Storing the inputs and the corresponding outputs of Boolean functions is an essential operation during the processing of these functions using quantum computers. Implementing Boolean functions through data structures is crucial for efficient computation and storage optimization, as it allows for compact representation and the faster evaluation of logical operations [3,4]. This approach is fundamental in areas such as circuit design, database indexing, and algorithm optimization. Among various fields, one of the most important issues that have been observed is the junta problem. The junta

problem is one process of learning an unknown Boolean function based only on variables that are uniformly selected at random from samples. Junta variable estimation is one such concept brought out by data structures, machine learning, and computational learning theory. A junta function can be related to a Boolean function that depends only on a small subset of its input variables, or junta variables. Being able to estimate their count may lead to efficient data storage and processing. In many real-world datasets or functions, only some of the features are relevant for decision-making or computation. It is expected that only a small, unknown fraction of a long DNA sequence will influence a genetic trait in studying the relationship between the two using computational biology [5]. Hence, it is effective to conduct a dimensionality reduction in machine learning applications where identification and estimation focus on junta variables, concentrating on the relevant ones and disregarding the irrelevant variables [6]. This is also important for saving storage so that only the relevant variables are stored instead of the whole dataset during the learning process. In addition, the computational efficiency is such that performing a computation on a smaller subset of variables reduces the requirements in terms of time and resources [7]. It is also an effective sparse representation in that, instead of storing the entire input space—which has a size 2^n for Boolean functions—it can represent the function using only the junta variables. Therefore, an estimation of the junta variables is effective for compressing the data structure. Hence, junta variable identification would imply that it can achieve data structure compression by encoding only the relevant variables and their relationship. This is particularly useful in applications such as decision trees or hash tables, where storage efficiency is at a premium. Knowing the junta variables allows us, in a database or search algorithm, to index queries depending on the attributes, which reduces the search space, hence yielding an improved query performance.

An investigation of junta variables was begun. Ambainis et al. [8] came up with a quantum learning method that can solve the problem of group testing using $\sqrt{\delta^{-1}k}$ function operations, where δ is the error tolerance and k is the number of junta inputs. A quantum method applicable to any Boolean function was developed by El-Wazan et al. [9]. Still another black-box function combining two function operations was developed by them. The quantum-based approach here uses $O(2^{\frac{n}{2}})$ function operations to identify dependent variables with a probability of at least $\frac{2}{3}$. However, no exact quantum learning method for the k-junta problem exists. For the rare case of 2-junta, the Boolean function of n variables $f(x) = x_g x_h$, where $0 \leq g$ and $h \leq n-1$, was proposed by Floess et al. [10]. They provided a quantum algorithm that, for any t function operations, examines variables that are dependent with a probability of $1-(0.25)^t$. Chen [11] provided a quantum algorithm for the solution of 2-junta with certainty. The worst time for Chen's algorithm to find two variables that are dependent is $O(\log_2 n)$. Therefore, this work was extended by Chen in Ref. [12] to find the three dependent variables of the Boolean function $f(x) = x_g x_h x_k$, where $0 \leq g$ and $h, k \leq n-1$ with the time complexity $O(\log_2 n)$. He also proved that this explained algorithm cannot solve the problem of k-junta using a single uncomplemented product. Chen [13] proposed an exact quantum learning procedure to solve the 2-junta problem, involving an $O(\log_2 n)$ function operation in the worst case. Also, Chen [14] proposed an exact algorithm for quantum learning to find the two dependent variables in solving the 2-junta problem. Later, in solving the 3-junta problem, Chen [15] proposed the exact algorithm of quantum learning to identify the three dependent variables. This algorithm solves the 3-junta problem using a modified black-box function. The quantum algorithm requires $O(\log_2 n)$ function operations in the worst-case scenario. Recently, in 2024, F. Aljuaydi et al. [16] used Zidan's quantum computing model to figure out if a variable is a junta or not for any unknown oracles and unknown input states. They were able to beat the classical approach in terms of both the time and memory

cost. However, estimating the number of junta variables for a Boolean function that is represented via an unknown oracle with a low time cost and low memory space is still an open problem. Therefore, this paper discusses a novel quantum circuit and algorithm to tackle this open research problem. The proposed technique was realized empirically using both IBM's quantum computer simulator and IBM's real quantum computer.

This paper is organized as follows: Section 2 explains the problem statement. The methodology that was used to develop the quantum algorithm that solved the assumed problem statement is explained in Section 3. The developed algorithm and its quantum circuit are explained in Section 4. The practical results are discussed in Section 6. Finally, the main results are summarized in Section 7.

2. Problem Statement

A junta function is a function that depends only on a small subset of its input variables, say k variables, where $k \leq n$ and n is the total number of inputs for this function. The variables that f depends on are called the junta variables. The considered problem is as follows:

Given: An oracle U_f that comprises an unknown Boolean expression $f : \{0,1\}^n \rightarrow \{0,1\}$.

Promise: The Boolean expression provided by the oracle U_f can comprise logical AND operations, logical OR operations, logical NOT operations, and logical XOR operations.

Goal: Determine the number of junta variables k for the oracle U_f with an allowable error ϵ^2.

To address this problem statement, a new quantum computing algorithm was proposed, which was assumed to be implemented using fault-tolerant quantum computers [17,18]. This type of quantum computer is currently under development, ensuring that the coherence of every quantum state remains unchanged throughout the operation and that errors are mitigated through quantum techniques. Additionally, fault-tolerant quantum computers are being developed to achieve perfect fidelity in quantum measurements, meaning that the outcome of a measurement depends solely on the probability defined by the ideal projective measurement [19,20].

3. Methodology: Zidan's Quantum Computing Model

The quantum computation model based on the degree of entanglement in the computational basis was proposed by Zidan [21,22]. This model operates by integrating the power of the quantum superposition output state, which results from operating oracles on a set of input qubits, and the initial state of the output qubit. The state of the output qubit is then handled using the degree of entanglement in the computational basis with the quantum Mz operator. This operator was proposed to quantify the degree of entanglement between two qubits [22]. The quantum circuit for this operator is depicted in Figure 1, and it is crucial for determining the number of variables on which the functions used for the k-junta testing depend.

The Mz operator is used in the proposed algorithm to solve the assumed problem statement. This operator receives two decoupled versions of the qubit $|l\rangle$. The normalized state of each version is $|l\rangle = q_1|0\rangle + q_4|1\rangle = \sqrt{\frac{t_1}{N}}|0\rangle + \sqrt{\frac{t_2}{N}}|1\rangle$, where $t_1 + t_2 = N = 2^n$. There are also two extra qubits, with each of them called $|h\rangle$, and they are initialized in the state $|0\rangle^{\otimes 2}$. Then, the M_z circuit applies two CNOT gates between each pair of the control qubit $|l\rangle$ and the target qubit $|h\rangle$, as shown in Figure 1, leaving each pair of these two qubits in the normalized state explained by Equation (1).

$$|l\ h\rangle = \sqrt{\frac{t_1}{2^n}}|00\rangle + \sqrt{\frac{t_2}{2^n}}|11\rangle. \quad (1)$$

Then, the state of the system that contains the two versions of the normalized state $|l\ h\rangle$ is described mathematically using Equation (2).

$$\begin{aligned} |\psi\rangle = & \quad |l\ h\rangle \otimes |l\ h\rangle \\ & = q_1^2|0000\rangle + q_1 q_4|0011\rangle + q_4 q_1|1100\rangle + q_4^2|1111\rangle, \end{aligned} \tag{2}$$

where $q_1 = \sqrt{\frac{t_1}{2^n}}$ and $q_4 = \sqrt{\frac{t_2}{2^n}}$. It is evident from Equation (2) that the value of q_4 can be estimated by Equation (3).

$$\begin{aligned} q_4^2 &= \sqrt{P_{1111}} \\ \sqrt{P_{1111}} &= \tfrac{t_2}{2^n}, \end{aligned} \tag{3}$$

where P_{1111} is the success probability of the basis state $|1111\rangle$. Thus, the value of t_2 can be calculated using the quantum computer via Equation (4).

$$t_2 = 2^n \times \sqrt{P_{1111}}. \tag{4}$$

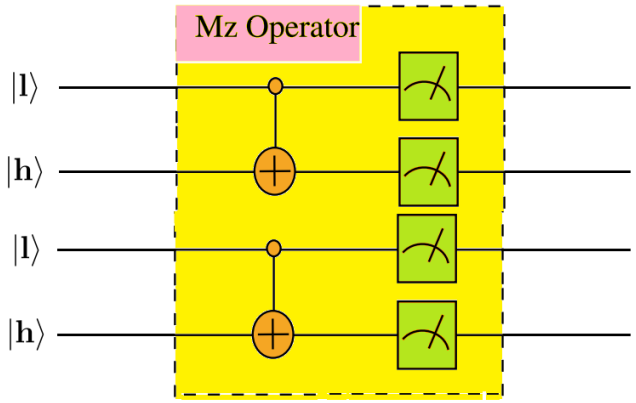


Figure 1. The components of the *Mz* operator.

4. The Proposed Quantum Algorithm and Analysis

4.1. Proposed Quantum Algorithm for Estimating the Number of K-Junta Variables

4.2. Analysis of the Proposed Algorithm

To determine the minimum number of variables k depended on by an oracle U_f that implements an unknown Boolean function, as explained in the problem statement assumed in Section 2, we propose Algorithm 1 to handle this problem. The quantum circuit of this suggested algorithm is illustrated in Figure 2. Here, the detailed steps of this algorithm are extensively discussed. In the first step, a Hadamard gate H is applied to each qubit in the register $|\chi\rangle$, generating all 2^n (N) possible rows of the truth table, which enables the identification of the number of variables on which the function depends ($0 < k < n$). Consequently, the system reaches the state $|\chi_1\rangle = H^{\otimes n} = \sum_{x=0}^{2^n-1} v_i|x\rangle$, where $v_i = 2^{\frac{-n}{2}}$. This operation creates a superposition of all possible inputs for the oracles U_f in Step 3. As a result of this superposition, the proposed quantum algorithm can query these oracles in constant time to determine t_2; this supports the achievement of the algorithm's goal to determine all function-dependent variables(k) in Step 7. In Step 2, the input consists of two registers: an n-qubit register $|\chi\rangle = |0\rangle^{\otimes n}$, which encodes the n Boolean variables representing the propositions of the oracles U_f, and a two-qubit register $|lh\rangle$. Initially, the system in the state $|\chi\rangle = |0\rangle^{\otimes n}|00\rangle$ is input based on the number of variables n of the function and the two ancillary qubits $|lh\rangle$. Step 3 generates the output of the unknown

Boolean function on the qubit $|l\rangle$; the oracle U_f is on the first $n+1$ qubit $|\chi\rangle \otimes |l\rangle$ of the systems $|\chi_2\rangle$ as follows:

$$|\chi_2\rangle = (U_f \otimes I^{\otimes 2})|\chi_1\rangle = \sum_i v_i |x\rangle |f(x)\rangle |0\rangle.$$

Algorithm 1 Proposed quantum algorithm to verify k-junta

Step 1: Apply n Hadamard gates on the state $|0\rangle^n$ to establish the superposed form as follows:

$$|\chi\rangle = H^{\otimes n} = \sum_{i=0}^{L-1} v_i |x\rangle, \text{where } v_i = 2^{\frac{-n}{2}},\ N = 2^n.$$

Step 2: Initialize a closed quantum system composed of the register $|\chi\rangle$ and two ancillary $|lh\rangle = |0\rangle^{\otimes 2}$; the state of this system is described by Equation (5).

$$|\chi_0^{S_1}\rangle = |\chi\rangle \otimes |lh\rangle = |\chi\rangle \otimes |00\rangle = \sum_i v_i |x\rangle |00\rangle. \tag{5}$$

Step 3: Apply the oracle $\mathcal{U}_f$ to compute the output of the Boolean function on the ancillary qubit $|l\rangle$. The new state becomes

$$|\chi_1^{S_1}\rangle = (I^{\otimes n} \otimes \mathcal{U}_f \otimes I)|\chi_0^{S_1}\rangle = \sum_i v_i |x\rangle |f(x)\rangle |0\rangle. \tag{6}$$

Step 4: Apply n quantum negation gates X to all qubits within the register $|\chi\rangle$, as follows:

$$\begin{aligned} |\chi_2^{S_1}\rangle = \quad & (X^{\otimes n} \otimes I^{\otimes 2})|\chi_1^{S_1}\rangle \\ & = \textstyle\sum_i v_i |x'\rangle |f(x)\rangle |0\rangle, \\ & \text{where } x' = (\neg x_0, \neg x_1, \ldots, \neg x_{n-1}). \end{aligned} \tag{7}$$

Step 5: Reapply the oracle $\mathcal{U}_f$ on the register $|\chi\rangle$ and the ancillary qubit $|l\rangle$, yielding

$$|\chi_3^{S_1}\rangle = (I^{\otimes n} \otimes \mathcal{U}_f \otimes I)|\chi_2^{S_1}\rangle = \sum_i v_i |x'\rangle |f(x) \oplus f(x')\rangle |0\rangle. \tag{8}$$

Step 6: Repeat the process described in Steps 1–6 to create a decoupled replica of the system. This operation does not violate the no-cloning theorem [23], as it creates an independent system. The combined state is

$$|\chi_4\rangle = |\chi_3^{S_1}\rangle \otimes |\chi_3^{S_2}\rangle. \tag{9}$$

Step 7: Perform the operator $\mathcal{M}_z$ on the qubits $|lh\rangle \otimes |lh\rangle$ and estimate the number of basis states t_2 that satisfy $f(y) = 1$ using Equation (4). The number of junta variables k is determined according to the following conditions:

(a) If $t_2 = 0$, then
$k = 0$.

(b) If $t_2 > 0$, then the number of junta variables is determined by Equation (10).

$$k = 1 - \frac{1}{2}\log_2 P_{1111}. \tag{10}$$

Step 4 applies n quantum negation gates X to all qubits within the register $|\chi\rangle$ to check whether there are variables that can change the result of the oracle U_f when applying the oracle again.

$$\begin{aligned} |\chi_2^{S_1}\rangle = \quad & (X^{\otimes n} \otimes I^{\otimes 2})|\chi_1^{S_1}\rangle \\ & = \textstyle\sum_i v_i |x'\rangle |f(x)\rangle |0\rangle, \\ & \text{where } x' = (\neg x_0, \neg x_1, \ldots, \neg x_{n-1}). \end{aligned} \tag{11}$$

Step 5 involves reapplying the oracle U_f to obtain $f(x) \oplus f(x')$ on the qubit $|l\rangle$, which has the initial state $|0\rangle$, so that we can compare after changing the output from the X gate in Step 4. This can be represented as follows:

$$|\chi_3^{S_1}\rangle = (I^{\otimes n} \otimes \mathcal{U}_f \otimes I)|\chi_2^{S_1}\rangle = \sum_i v_i |x'\rangle |f(x) \oplus f(x')\rangle |0\rangle.$$

Step 6 reiterates Steps 1–5 to generate an additional replica $|\chi_3\rangle$ of the system. This duplication is necessary because the M_z operator operates on two copies of $|lh\rangle$ to extract the degree of entanglement between these two qubits. At this stage, both copies of the qubits $|l\rangle$ and $|h\rangle$ are available, and the overall state of the system is given by

$$|\chi_4\rangle = |\chi_3^{S_1}\rangle \otimes |\chi_3^{S_2}\rangle.$$

The final step significantly contributes to determining the number of variables that the function depends on (k) when calculating t_2. After the M_z operator, Step 7 involves entanglement creation, firstly by applying the CNOT gate on $|lh\rangle$ and $|lh\rangle$ in another replica, and secondly by determining the degree of entanglement between these, $|\chi\rangle = |lh\rangle \otimes |lh\rangle$, as in Equation (2). Consequently, there are two possible cases. Case (a): When calculating t_2 and finding $t_2 = 0$, the value of k can be determined as indicated in the proposed algorithm. In this case, $k = 0$. Case (b): In this case, $t_2 > 0$; the value of t_2 used to determine k, as shown in the proposed algorithm, is determined using Equation (10).

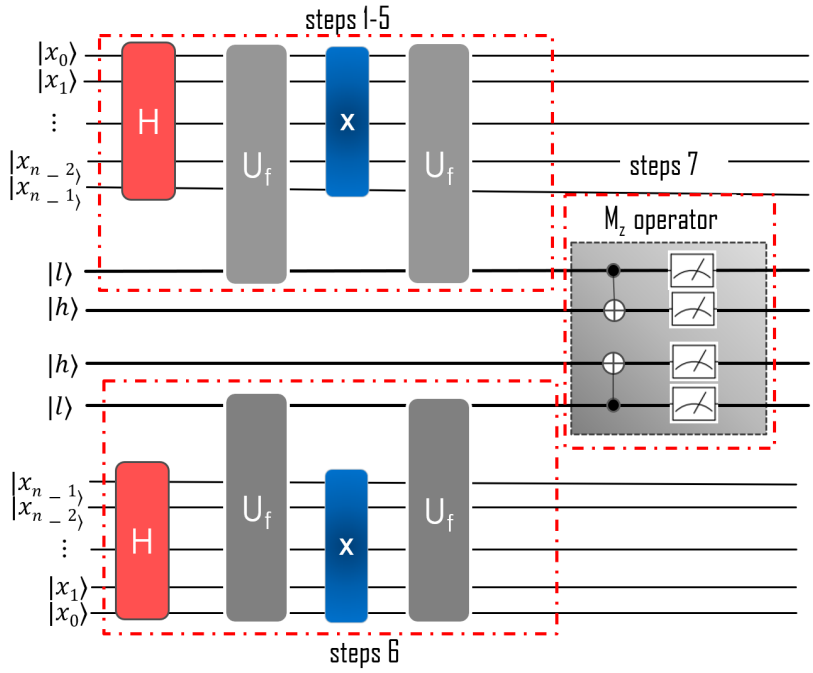


Figure 2. The detailed elementary quantum circuit of the proposed approach classifies variables of unknown logical functions that receive unknown quantum input states into k-junta classes.

In further explanation, the success probability P_{1111} of measuring the basis state $|1\rangle^{\otimes 4}$ for the qubits $|lhlh\rangle$ after applying the Mz operator is related to the number of junta variables k from Equation (10). According to Equation (4), $t_2 = 2^n \times \sqrt{P_{1111}}$, where t_2 represents the number of inputs for which $f(x) \neq f(x')$. A Boolean function that depends on exactly k-junta variables is independent of the remaining $n - k$. Additionally, the negation operation performed by Step 4 creates pairs of inputs (x, x') to check the

function f dependency on variable changes. Therefore, the number of input pairs (x, x') that experience a differing output is determined by Equation (12).

$$t_2 = 2^{n-(k-1)}. \tag{12}$$

By equating Equations (4) and (12), we obtain the term $2^{n-(k-1)} = 2^n \times \sqrt{P_{1111}}$, which yields $2^{-(k-1)} = \sqrt{P_{1111}}$. Therefore, squaring both sides yields $P_{1111} = 2^{-2(k-1)}$. By taking the logarithm and solving for k, we obtain $k = 1 - \frac{1}{2}\log_2 P_{1111}$.

5. Time Complexity Analysis

Here, we analyze the time complexity of the proposed approach based on the standard oracle model of quantum computation, known as quantum query complexity theory (see Refs. [24–26]). In this model, an oracle is treated as a black-box unitary operation, and its application is counted as a single computational step (see Refs. [24–26]), independent of the internal workings of the oracle U_f at hand. This establishes a clear separation between the query complexity (the number of oracle calls) and the computational cost of implementing the Boolean function that is hidden within the black box of the oracle U_f in certain hardware [24–26]. According to Step 7 of the proposed algorithm, the number of junta variables k of an oracle U_f is determined as follows: $k = 0$ only if $t = 0$; otherwise, it is given by $k = 1 - \frac{1}{2}\log_2 P_{1111}$, where P_{1111} is the success probability of the basis state $|1111\rangle$ for the qubits $|lhlh\rangle$ obtained from the Mz operator. It should be noted that P_{1111} is estimated as the number of successes when estimating the basis state $|1111\rangle$ from the Mz operator after applying Steps 1–6 in the proposed algorithm M times over the number of trials M. Here, it should be noted that the oracle, U_f, is recalled twice in each trial, since M is a limited number of occurrences (shots), and according to Equation (2), it follows the binomial distribution. Therefore, the variance of p_{1111} is given by $Var(p_{1111}) = \frac{p_{1111}(1-p_{1111})}{M}$. Consequently, the standard error of p_{1111} is $\epsilon_{p_{1111}} = \sqrt{\frac{p_{1111}(1-p_{1111})}{M}}$. The number of junta variables $k = 1 - \frac{1}{2}\log_2 P_{1111}$ can be written as $k = 1 - \frac{\ln P_{1111}}{2\ln 2}$ after re-expressing the term $\log_2 P_{1111}$ as $\log_2 P_{1111} = \frac{\ln P_{1111}}{\ln 2}$. By differentiating the term $k = 1 - \frac{\ln P_{1111}}{2\ln 2}$ with respect to p_{1111}, we obtain $\frac{dk}{dp_{1111}} = -\frac{1}{2p_{1111}\ln 2}$. For a small error in p_{1111}, the approximate standard error in k can be expressed as $\epsilon \approx |\frac{dk}{dp_{1111}}| \times \epsilon_{p_{1111}} = \frac{1}{2\ln 2}\sqrt{\frac{1-p_{1111}}{Mp_{1111}}}$. Consequently, the number of times required to recall the oracle U_f for estimating k with an allowable error using the proposed algorithm is given by $\epsilon \propto \frac{1}{\sqrt{M}} \implies M \propto \frac{1}{\epsilon^2}$. Since the time complexity is $O(M)$, this renders the time complexity of the proposed approach as $O(\frac{1}{\epsilon^2})$.

6. Experimental Realization of the Proposed Algorithm

The proposed algorithm was realized practically to examine the number of junta variables k by implementing its quantum circuit on IBM's quantum computer simulator and IBM's real quantum computer. Here, nine experiments were conducted for nine different oracles. Each oracle was designed with a distinct number of input variables to evaluate the effect of the input dimensionality on the performance of the models, such that each oracle was designed to be known only to the experiment's designer and hidden from anyone else. The quantum circuit of the proposed algorithm in each experiment was carried out with 20,000 shots. The statistical fidelities of both the simulated results and the real quantum computer results, in each experiment, were calculated as follows: $F_{\text{sim}} = \sum_{j=0}^{2^n-1}\sqrt{p_j^{sim}p_j^{th}}$ and $F_{\text{rc}} = \sum_{j=0}^{2^n-1}\sqrt{p_j^{rc}p_j^{th}}$, where n is the number of qubits and p_j^{sim}, p_j^{th}, and p_j^{rc} are the simulation probabilities, theoretical probabilities, and real quantum computer probabilities, respectively.

Here, the steps for creating the oracles $\{U_f\}$ in the conducted experiments are detailed. To build the quantum circuit as an oracle to encode a given Boolean function, the following steps were executed: (i) Create the truth table of the Boolean function at hand, and then retrieve from this truth table the input that produces a value of 1 when evaluating the Boolean function in the truth table. (ii) Take these rederived inputs and combine them in the Boolean expression such that each input term is related to the other terms by a logical OR operator. (iii) Replace each OR operation with an XOR operation using Reed–Muller form, and simplify the obtained Boolean expression [27]. (iv) Convert the final formula of the Boolean expression into a quantum circuit of $n+1$ qubits, where n represents the number of input variables and the extra qubit is for the function's output. Then, each term in the Boolean expression is represented as a CNOT gate, or Toffoli gate, in this quantum circuit of the oracle [27].

In the first experiment, the first oracle encoded the Boolean function $f = 0$. The results of this experiment are presented in Figure 3. It can be observed that the probability of P_{0000} was equal to 1; this means that $P_{1111} = 0$ in both the theoretical calculations and the simulation results. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = k_{simulation} = 0$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis states $|0000\rangle$ and $|1111\rangle$ were $P_{0000} = 0.69349$ and $P_{1111} = 0.00007$. So, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 0$, which aligns with the theoretical value. The reason that there was no difference between the value of k for the real quantum computer and the theoretical result, despite the fact that the real quantum computer is noisy, is because the oracle encoded a simple function $f = 0$. The statistical fidelities of this experiment were $F_{\text{simulation}} = 1$ and $F_{\text{real quantum computer}} = 0.83276047$.

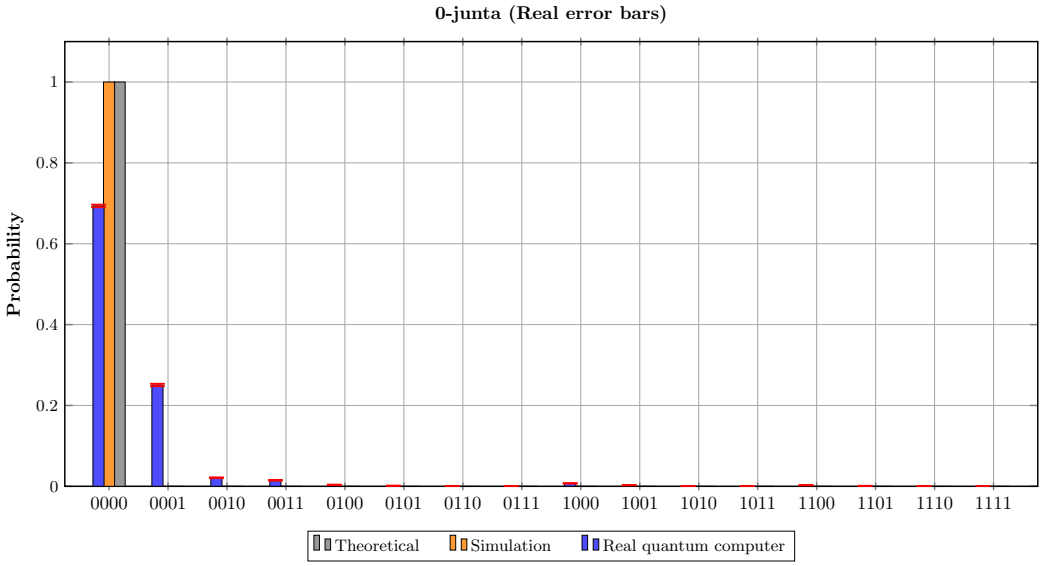


Figure 3. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = 0$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

In the second experiment, the second oracle encoded the Boolean function $f = x_0$. The results of this experiment are presented in Figure 4. It can be observed that the

probability of P_{1111} was equal to 1; this means that $P_{0000} = 0$ in both the theoretical calculations and the simulation results. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = k_{simulation} = 1$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis states $|0000\rangle$ and $|1111\rangle$ were $P_{0000} = 0.000235$ and $P_{1111} = 0.826655$. So, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 1.13732137 \approx 1$, which aligns with the theoretical value. The statistical fidelities of this experiment were $F_{\text{simulation}} \approx 1$ and $F_{\text{real quantum computer}} = 0.909205697$.

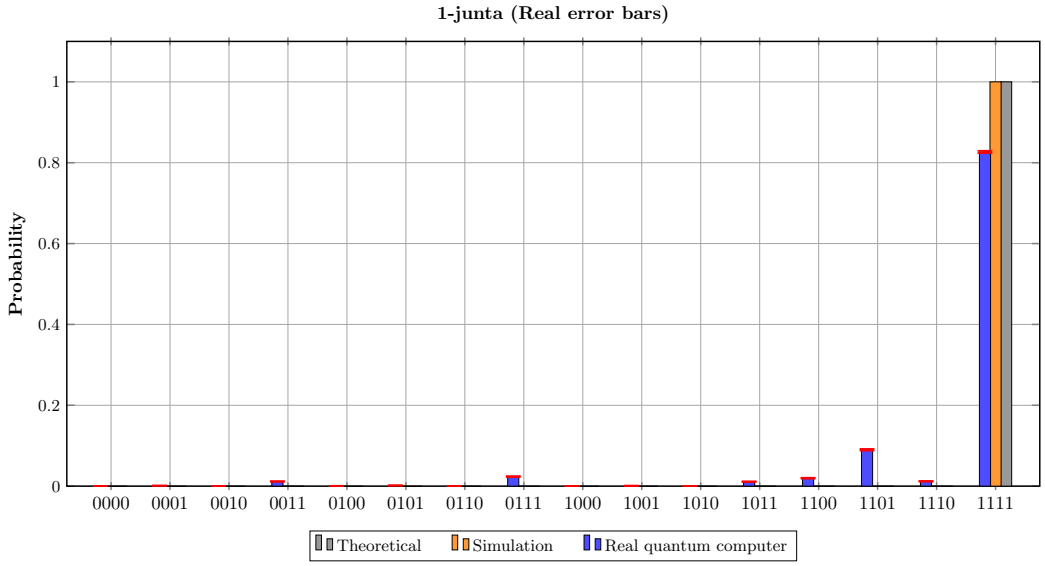


Figure 4. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

In the third experiment, the third oracle encoded the Boolean function $f = x_0 \cdot x_1$. The results of this experiment are presented in Figure 5. It can be observed that the probabilities $P_{0000} = 0.25$, $P_{0011} = 0.25$, $P_{1100} = 0.25$, and $P_{1111} = 0.25$ in the theoretical calculations and in the simulation results were approximately $P_{0000} = 0.250485$, $P_{0011} = 0.25118$, $P_{1100} = 0.249435$, and $P_{1111} = 0.2489$. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = 2$ and $k_{simulation} = 2.003180932$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis state were $P_{0000} = 0.223765$, $P_{0011} = 0.22652$, $P_{1100} = 0.201295$, and $P_{1111} = 0.1997$. So, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 2.162046881 \approx 2$, which aligns with the theoretical value. The statistical fidelities of this experiment were $F_{\text{simulation}} = 0.999998422$ and $F_{\text{real quantum computer}} = 0.9222582$.

In the fourth experiment, the oracle encoded the Boolean function $f = x_0 \cdot x_1 \cdot x_2$. The results of this experiment are presented in Figure 6. It can be observed that the probabilities $P_{0000} = 0.5625$, $P_{0011} = 0.1875$, $P_{1100} = 0.1875$, and $P_{1111} = 0.0625$ in the theoretical calculations and in the simulation results were approximately $P_{0000} = 0.565135$, $P_{0011} = 0.186505$, $P_{1100} = 0.18618$, and $P_{1111} = 0.06218$. Consequently, both theoretically and according to the

simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = k_{simulation} = 3$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis states were $P_{0000} = 0.342845$, $P_{0011} = 0.19909$, $P_{1100} = 0.177885$, and $P_{1111} = 0.10174$. Consequently, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 2.648520548 \approx 3$. The statistical fidelities of this experiment were $F_{\text{simulation}} = 0.999996428$ and $F_{\text{real quantum computer}} = 0.894726375$.

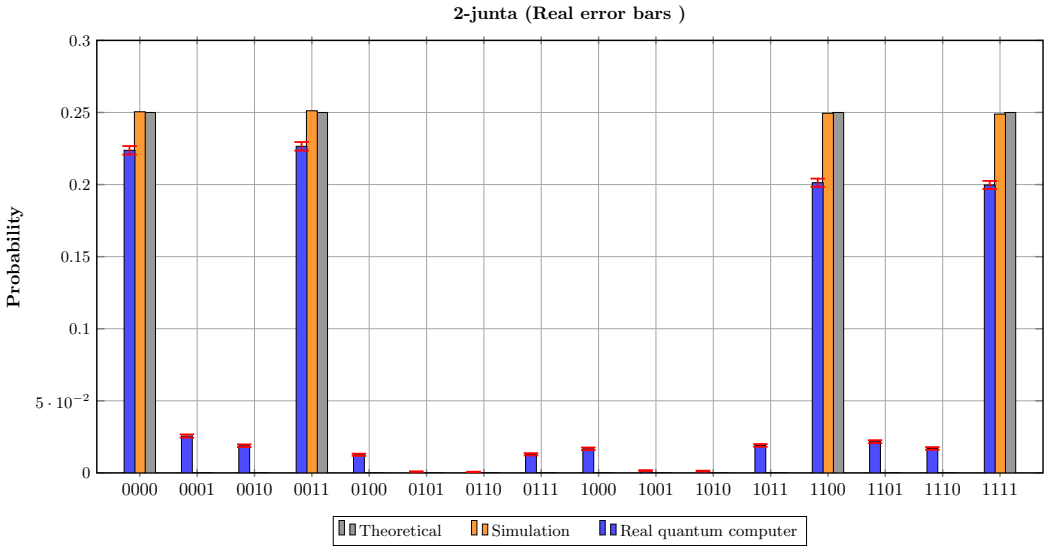


Figure 5. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0 \cdot x_1$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

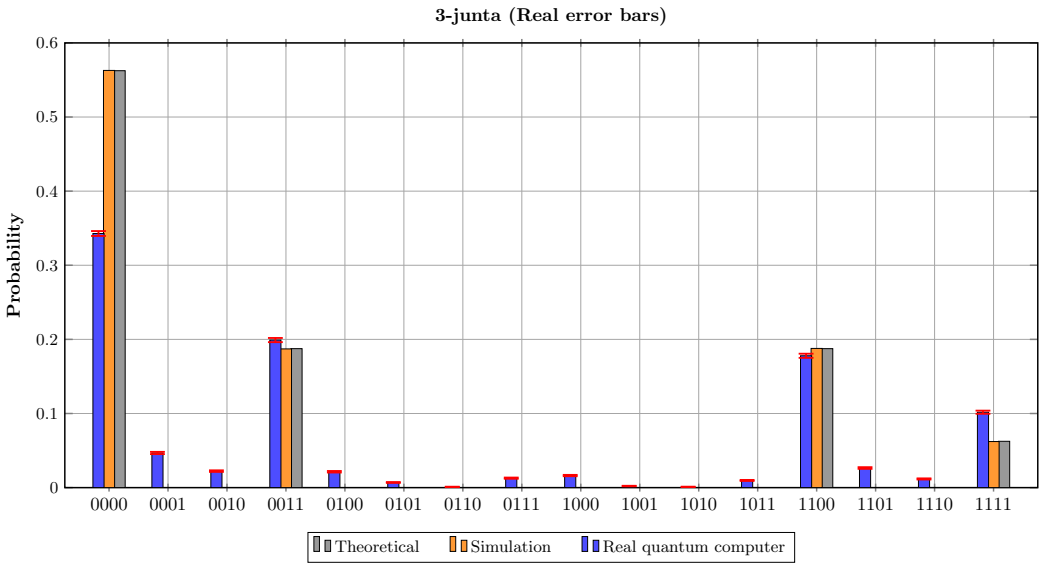


Figure 6. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0 \cdot x_1 \cdot x_2$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

In the fifth experiment, the oracle encoded the Boolean function $f = x_0 \cdot x_1 \cdot x_2 \cdot x_3$. The results of this experiment are presented in Figure 7. It can be observed that the probabilities $P_{0000} = 0.76564$, $P_{0011} = 0.10955$, $P_{1100} = 0.10955$, and $P_{1111} = 0.01526$ in the theoretical calculations and in the simulation results were approximately $P_{0000} = 0.76574$, $P_{0011} = 0.10955$, $P_{1100} = 0.109445$, and $P_{1111} = 0.015265$. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = 4.017050614$ and $k_{simulation} = 4.017050614$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis state were $P_{0000} = 0.229305$, $P_{0011} = 0.1996$, $P_{1100} = 0.199415$, and $P_{1111} = 0.172925$. It is clear that, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 2.265890819$. The statistical fidelities of this experiment were $F_{\text{simulation}} = 1$ and $F_{\text{real quantum computer}} = 0.766050271$.

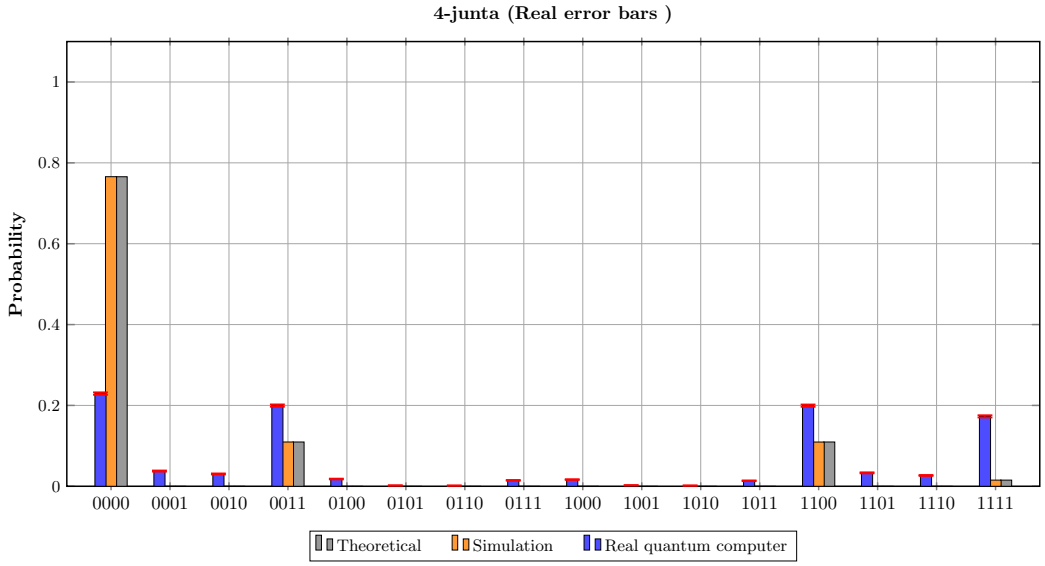


Figure 7. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0 \cdot x_1 \cdot x_2 \cdot x_3$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

In the sixth experiment, the oracle encoded the Boolean function $f = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4$. The results of this experiment are presented in Figure 8. It can be observed that the probabilities $P_{0000} = 0.87862$, $P_{0011} = 0.0589$, $P_{1100} = 0.0589$, and $P_{1111} = 0.00358$ in the theoretical calculations and in the simulation results were approximately $P_{0000} = 0.87862$, $P_{0011} = 0.05894$, $P_{1100} = 0.05866$, and $P_{1111} = 0.00378$. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = 5.062912349$ and $k_{simulation} = 5.062912349$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis state were $P_{0000} = 0.212035$, $P_{0011} = 0.19152$, $P_{1100} = 0.203365$, and $P_{1111} = 0.185055$. Hence, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 2.216986989$. The statistical fidelities of this experiment were $F_{\text{simulation}} = 0.999998515$ and $F_{\text{real quantum computer}} = 0.673016544$.

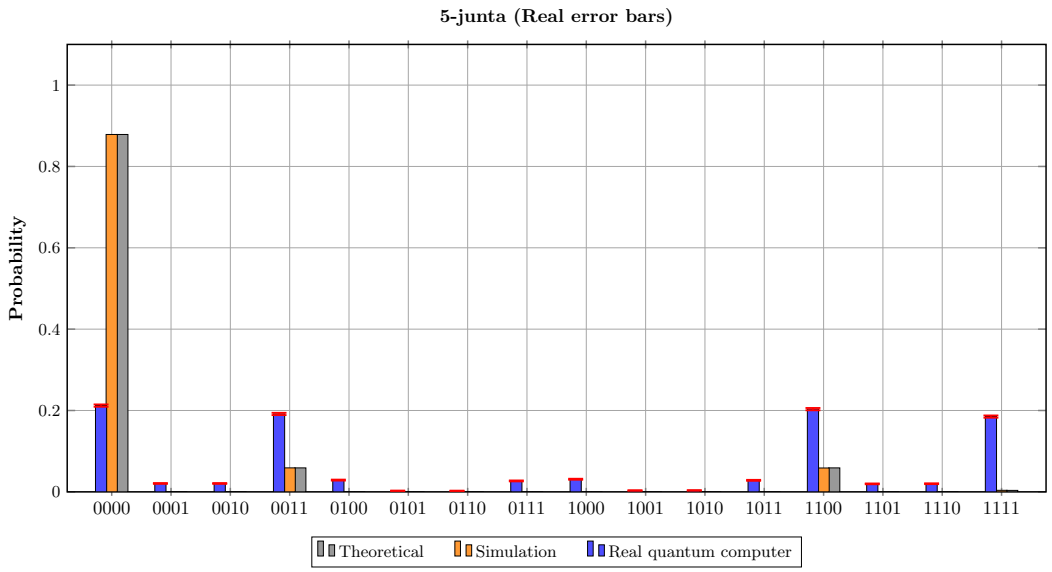


Figure 8. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

In the seventh experiment, the oracle encoded the Boolean function $f = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5$. The results of this experiment are presented in Figure 9. It can be observed that the probabilities of $P_{0000} = 0.939, P_{0011} = 0.03, P_{1100} = 0.03$, and $P_{1111} = 0.001$ in the theoretical calculations and in the simulation results were approximately $P_{0000} = 0.93905$, $P_{0011} = 0.030275, P_{1100} = 0.029565$, and $P_{1111} = 0.00111$. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = 5.907612304$ and $k_{simulation} = 5.982892142$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis state were $P_{0000} = 0.22652$, $P_{0011} = 0.18867, P_{1100} = 0.184445$, and $P_{1111} = 0.152085$. According to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 2.358525113$. The statistical fidelities of this experiment were $F_{\text{simulation}} = 0.999997457$ and $F_{\text{real quantum computer}} = 0.623148976$.

In the eighth experiment, the oracle encoded the Boolean function $f = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6$. The results of this experiment are presented in Figure 10. It can be observed that the probabilities $P_{0000} = 0.9698, P_{0011} = 0.015, P_{1100} = 0.015$, and $P_{1111} = 0.0002$ in the theoretical calculations and in the simulation results were approximately $P_{0000} = 0.969405$, $P_{0011} = 0.015005, P_{1100} = 0.01538$, and $P_{1111} = 0.00021$. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = 7.108661526$ and $k_{simulation} = 7.14385619$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis state were $P_{0000} = 0.236765$, $P_{0011} = 0.22322, P_{1100} = 0.196125$, and $P_{1111} = 0.186545$. Thus, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 2.211202202$. The statistical fidelities of this experiment were $F_{\text{simulation}} = 0.99999873$ and $F_{\text{real quantum computer}} = 0.597392938$.

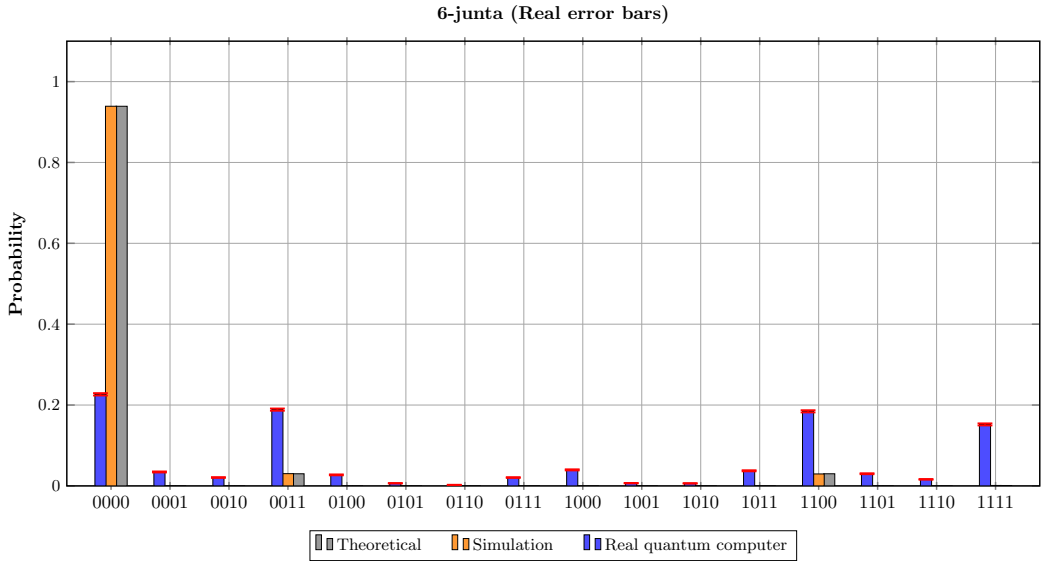


Figure 9. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

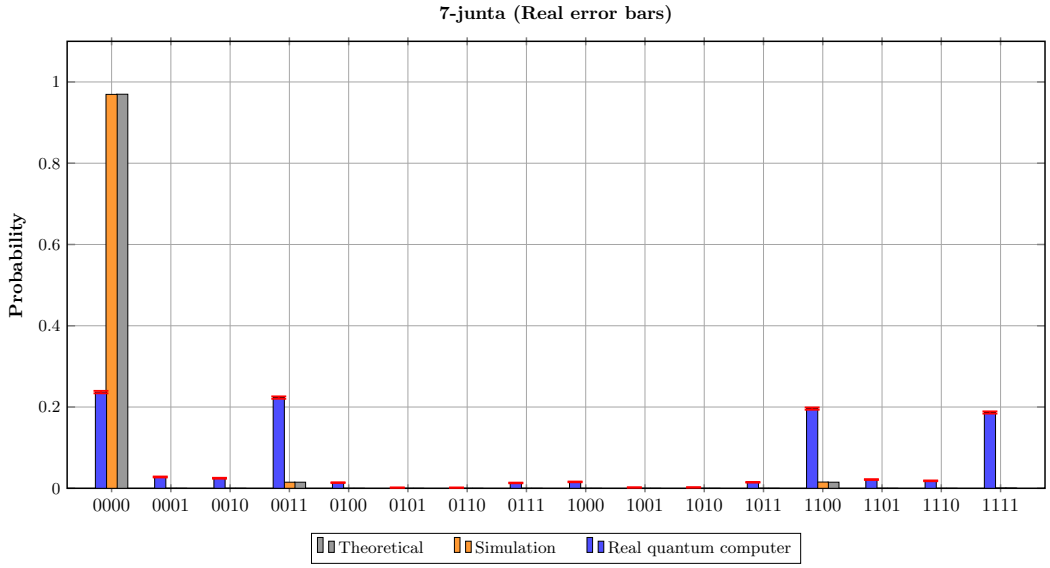


Figure 10. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

Finally, in the ninth experiment, the oracle encoded the Boolean function $f = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7$. The results of this experiment are presented in Figure 11. It can be observed that the probabilities of $P_{0000} = 0.984415$, $P_{0011} = 0.00777$, $P_{1100} = 0.00777$, and $P_{1111} = 0.000045$ in the theoretical calculations and in the simulation results were approximately $P_{0000} = 0.9844$, $P_{0011} = 0.00777$, $P_{1100} = 0.007785$, and $P_{1111} = 0.000045$. Consequently, both theoretically and according to the simulation result, when estimating k for this oracle using Step 7 of the proposed algorithm, $k_{theoretical} = k_{simulation} = 8.219857736$. In contrast, when the quantum circuit of the proposed algorithm was implemented on the real quantum computer for the first oracle, the probabilities of the basis state

were $P_{0000} = 0.28032$, $P_{0011} = 0.224165$, $P_{1100} = 0.20699$, and $P_{1111} = 0.16026$. So, according to Step 7 in the proposed algorithm, the number of junta variables, which was estimated when implementing the algorithm on the real quantum computer, was $k_{\text{real quantum computer}} = 2.320756857$. The statistical fidelities of this experiment were $F_{\text{simulation}} = 1$ and $F_{\text{real quantum computer}} = 0.60983424$. Overall, it is evident that both the theoretical and simulation results for estimating the number of junta variables were aligned in all experiments. This alignment occurred because the simulation results represent the outcomes of implementing the proposed approach in futuristic, fault-tolerant real quantum computers, which are free from noise and mitigate inherent errors [17,18]. On the other hand, in the first experiment, where the number of junta variables was 0, the theoretical, simulation, and real quantum computer results for estimating the number of junta variables were also aligned. Additionally, the difference between the theoretical and simulation results on one side and the real quantum computer results on the other side for estimating the number of junta variables was small and can be considered negligible in experiments 2–4, where the number of junta variables was 1, 2, and 3, respectively. Nevertheless, the gap between the theoretical and simulation results on one side and the real quantum computer results on the other side for estimating the number of junta variables increased in Experiments 5–9, where the number of junta variables ranged from 4 to 8. The difference between the values of the junta variables from the real quantum computer and the theoretical results in these experiments can be attributed to two factors: (i) Current real quantum computers are noisy devices and are not resistant to noise, which is why they are referred to as noisy intermediate-scale quantum (NISQ) computers [19,20]. Nevertheless, several research efforts are striving to develop fault-tolerant quantum computers by 2029. The number of junta variables in the oracles used in Experiments 5–9 increased from 4 to 9. This means that these oracles had complex internal structures in terms of CNOT gates and multi-controlled gates, which led to the accumulation of errors during the implementation of these gates in the current IBM's real quantum computers. Additionally, the decoherence times scaled with the circuit depth, which is related to the number of gates in the entire quantum circuit, explaining the observed divergence.

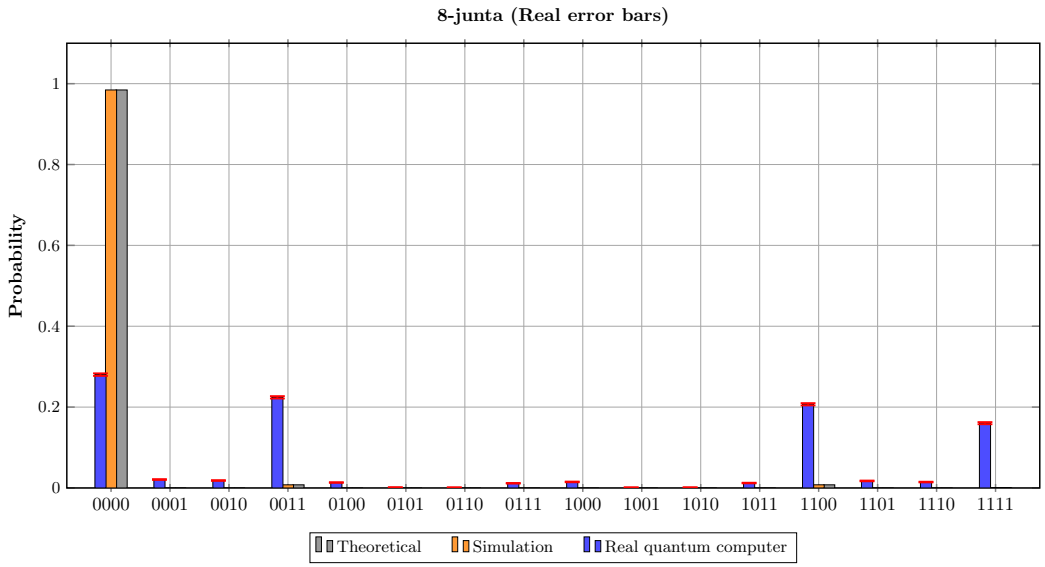


Figure 11. The experimental results of realizing the proposed quantum algorithm using both the quantum computer simulator and the real quantum computer were compared to the theoretical results to examine an oracle that implements a Boolean function in the form $f(x_0, x_1, \ldots, x_7) = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7$. The red bars in the blue histograms show the standard deviation of obtaining the basis states, presented in terms of implementing the proposed algorithm on IBM's real quantum computer.

Table 1 explains the time cost of the proposed approach for estimating the number of k-junta variables for an unknown oracle compared to the classical approach. The time complexity for the classical approach is 2^{n+1} and the required memory space in this approach is 2^{n+1} as well (the detailed proof is investigated in [16]). It has been proven that the time complexity of handling the quantum computing problems using Zidan's quantum computing model based on the Mz operator is $O(\frac{1}{\epsilon^2})$. Subsequently, the time cost for handling the proposed problem statement that is explained in Section 2 using the suggested algorithm explained in Section 4 is $O(\frac{1}{\epsilon^2})$ (the detailed proof is investigated in [16,22,28]), where ϵ is a predefined error. This means that the developed algorithm is independent of the number of inputs n. It also means that the developed quantum algorithm that solves the problem statement explained in Section 2 in constant time is compared to the classical approach, which requires exponential time. However, it is worth noting that, while the classical approach handles the problem in exponential time, it is deterministic. Conversely, the developed approach solves the same problem in constant time with allowable error ϵ. Furthermore, the proposed quantum approach requires $n+2$ qubits in each version of the quantum system, which has two versions (see Section 4). Therefore, the total memory space that is required for the proposed quantum approach is $2n+4$. Hence, it is clear that the proposed algorithm achieves quantum supremacy in terms of memory space and time complexity.

Table 1. Comparing the time cost and memory cost of estimating the number of junta variables using the developed quantum approach and its classical counterpart.

Approach Name	Proposed Quantum Algorithm	Classical Algorithm
Memory Space:	$2n+4$.	2^{n+1}.
Time Cost:	$O(\frac{1}{\epsilon^2})$.	2^{n+1}.

7. Conclusions

This study discusses a novel quantum method for estimating the minimum number of k-junta inputs for a given unknown Boolean function that has a set of arbitrary inputs. The proposed method achieved an exponential speed-up over classical techniques, operating with a time complexity of $2\epsilon^{-2}$ as the number of Boolean variables increased. The experimental results conducted using IBM's real quantum computer and the Qiskit simulator to practically realize the suggested algorithm showed that this challenge is intractable for both classical computers and advanced quantum methods, yet it becomes feasible with the proposed approach. The proposed approach was demonstrated to solve the same problem when the oracle is unknown, the inputs are incomplete, and there is a weighted superposition of some basis states. In future work, we will develop the quantum circuit of the proposed algorithm to mitigate errors in noisy quantum computers and to perform effectively using noisy intermediate-scale quantum (NISQ) computers.

Author Contributions: Conceptualization, S.S.A.; methodology, A.A. and S.A.; writing—original draft, S.A.; writing—review and editing, M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Prince Sattam bin Abdulaziz University through the project number (PSAU/2024/01/29770).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2024/01/29770).

Conflicts of Interest: All authors declare that there are no conflicts of interest.

References

1. Zidan, M.; Abdel-Aty, A.-H.; Khalil, A.; Abdel-Aty, M.; Eleuch, H. A novel efficient quantum random access memory. *IEEE Access* **2021**, *9*, 151775–151780. [CrossRef]
2. ElEuch, H.; Zidan, M.A.A.; Abdelaty, A.M.A.; Abdel-Aty, M.M.A.; Khalil, A. Quantum Random Access Memory System. U.S. Patent 11651266, 2023.
3. Bryant, R.E. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv. (CSUR)* **1992**, *24*, 293–318. [CrossRef]
4. Knuth, D. *The Art of Computer Programming*; Volume 4A: Combinatorial algorithms, Part 1; Addison-Wesley: Boston, MA, USA, 2011; ISBN 0321751043.
5. Mossel, E.; O'Donnell, R.; Servedio, R.P. Learning functions of k relevant variables. *J. Comput. Syst. Sci.* **2004**, *69*, 421–434. [CrossRef]
6. Lu, Z.Q. The elements of statistical learning: Data mining, inference, and prediction. *J. R. Stat. Soc. Ser. A* **2010**, *173*, 693–694.
7. Mossel, E.; O'Donnell, R.; Servedio, R.P. Learning juntas. In Proceedings of the 35th Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 9–11 June 2003; pp. 206–212.
8. Ambainis, A.; Belovs, A.; Regev, O.; de Wolf, R. Efficient quantum algorithms for (gapped) group testing and junta testing. In Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms, Arlington, VA, USA, 10–12 January 2016; pp. 903–922.
9. El-Wazan, K.; Younes, A.; Doma, S.B. A quantum algorithm for testing juntas in Boolean functions. *arXiv* **2017**, arXiv:1701.02143.
10. Floess, D.F.; Andersson, E.; Hillery, M. Quantum algorithms for testing Boolean functions. *arXiv* **2010**, arXiv:1006.1423. [CrossRef]
11. Chen, C.-Y. An exact quantum algorithm for testing Boolean functions with one un complemented product of two variables. *Quantum Inf. Process.* **2020**, *19*, 213. [CrossRef]
12. Chen, C.-Y. An exact quantum algorithm for testing 3-junta in boolean functions with one uncomplemented product. *Quantum Inf. Process.* **2021**, *20*, 36. [CrossRef]
13. Chen, C.-Y. An exact quantum algorithm for the 2-junta problem. *Int. J. Theor. Phys.* **2021**, *60*, 80–91. [CrossRef]
14. Chen, C.-Y. An exact quantum learning algorithm for the 2-junta problem in constant time. *Int. J. Theor. Phys.* **2022**, *61*, 212. [CrossRef]
15. Chen, C.-Y. An exact quantum logarithmic time algorithm for the 3-junta problem. *Quantum Inf. Process.* **2024**, *23*, 232. [CrossRef]
16. Aljuaydi, F.; AbdelAzim, S.; Darwish, M.M.; Zidan, M. A Quantum Algorithm for Boolean Functions Processing. *IEEE Access* **2024**, *12*, 164503–164519. [CrossRef]
17. Zhang, R.; Wang, G.; Johnson, P. Computing Ground State Properties with Early Fault-Tolerant Quantum Computers. *Quantum* **2022**, *6*, 761. [CrossRef]
18. Katabarwa, A.; Gratsea, K.; Caesura, A.; Johnson, P. Early Fault-Tolerant Quantum Computing. *PRX Quantum* **2024**, *5*, 020101. [CrossRef]
19. Ryan-Anderson, C.; Bohnet, J.G.; Lee, K.; Gresh, D.; Hankin, A.; Gaebler, J.P.; Francois, D.; Chernoguzov, A.; Lucchetti, D.; Brown, N.C.; et al. Realization of Real-Time Fault-Tolerant Quantum Error Correction. *Phys. Rev. X* **2021**, *11*, 041058. [CrossRef]
20. Sivak, V.V.; Eickbusch, A.; Royer, B.; Singh, S.; Tsioutsios, I.; Ganjam, S.; Miano, A.; Brock, B.L.; Ding, A.Z.; Frunzio, L.; et al. Real-time quantum error correction beyond break-even. *Nature* **2023**, *616*, 50–55. [CrossRef] [PubMed]
21. Zidan, M. A novel quantum computing model based on entanglement degree. *Mod. Phys. Lett. B* **2020**, *34*, 2050401. [CrossRef]
22. Zidan, M.; Hegazy, S.F.; Abdel-Aty, M.; Obayya, S.S.A. Rapid solution of logical equivalence problems by quantum computation algorithm. *Appl. Soft Comput.* **2023**, *132*, 109844. [CrossRef]
23. Wootters, W.K.; Zurek, W.H. The no-cloning theorem. *Phys. Today* **2009**, *62*, 76–77. [CrossRef]
24. Deutsch, D.; Jozsa, R. Rapid solutions of problems by quantum computation. *Proc. R. Soc. Lond. A* **1992**, *439*, 553–558.
25. Simon, D.R. On the Power of Quantum Computation. *SIAM J. Comput.* **1997**, *26*, 1474–1483. [CrossRef]
26. Grover, L.K. Quantum Mechanics Helps in Searching for a Needle in a Haystack. *Phys. Rev. Lett.* **1997**, *79*, 325. [CrossRef]
27. Barnum, H.; Bernstein, H.J.; Spector, L. Quantum circuits for OR and AND of ORs. *J. Phys. A Math. Gen.* **2000**, *33*, 8047–8057. [CrossRef]
28. Zidan, M.; Eisa, A.M.; Qasymeh, M.; Shoman, M.A.I. A Quantum Algorithm for System Specifications Verification. *IEEE Internet Things J.* **2024**, *early access*. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.