Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Quantum Approach for Market Basket Analysis on Qudit Systems

Semi O. Ceylan

osman.semi.ceylan@gmail.com

Çanakkale Onsekiz Mart University, Turkey

Sait Can Yücebaş

Çanakkale Onsekiz Mart University, Turkey

Research Article

Keywords: Quantum Computing, Qudit Systems, Market Basket Analysis, Grover Search Algorithm, Dicke States

Posted Date: March 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8269117/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Quantum Approach for Market Basket Analysis on Qudit Systems

Semi O. Ceylan[1, 2*] and Sait Can Yücebaş[1]

[1*]Department of Computer Engineering, Çanakkale Onsekiz Mart University, Canakkale, 17020, Türkiye.
[2]Department of Computer Engineering, Maltepe University, Istanbul, 34857, Türkiye.

*Corresponding author(s). E-mail(s): osman.semi.ceylan@gmail.com;
Contributing authors: can@comu.edu.tr;

Abstract

Quantum computing consists of a set of techniques based on the principles of quantum mechanics. Historically, quantum computing has been built around qubit systems featuring two discrete levels. However, qudit systems involving multiple levels appear to provide a promising solutions to certain difficulties related to noise and information capacity in quantum computing. This study investigate these advantages by extending the quantum approaches of market basket analysis (MBA) on qudit systems. Furthermore, we propose a novel method utilising an adaptive comparator that requires fewer quantum resources in assessing the minimum support threshold, made possible by qudit systems. The proposed quantum MBA method was simulated on an 8-level system for a database containing 9 items and 64 transactions. The experiment results show that the proposed method can perform MBA on the same database using 78,125% less quantum resources compared to existing methods. All simulations conducted in this study utilize the Cirq framework.

Keywords: Quantum Computing, Qudit Systems, Market Basket Analysis, Grover Search Algorithm, Dicke States

1 Introduction

Market Basket Analysis (MBA) is a data mining methodology to identify patterns of co-occurrence among items in transactional datasets, revealing relationships between products that are frequently purchased together [1]. MBA provides insights into customer purchasing behaviours, facilitating informed decision-making in various domains. For instance, in retail and e-commerce, MBA is instrumental in optimizing store layouts, designing targeted marketing campaigns, and developing recommendation systems to enhance cross-selling and up-selling strategies. MBA is also used in domains such as fraud detection to uncover patterns indicative of fraudulent activities [2], and in healthcare to analyse patient data and optimize medical services [3]. Furthermore, MBA extends to sectors like telecommunications and energy, where it aids in customer behaviour analysis, service optimization, and resource allocation [4]. This wide-ranging applicability underscores MBA's versatility in addressing sector-specific challenges beyond its traditional retail context. Regardless of the domain, various methods have been developed to make MBA analysis more accurate and resource-efficient.

The AIS algorithm [5] is one of the first methods proposed for MBA. It requires multiple scans of the database to identify sufficiently large item sets. During the initial scan, *1*-item itemsets are generated and their support counts are computed. Sets below a predefined threshold are discarded, while those above are retained as candidates. Subsequent scans expand candidate sets incrementally (e.g., to *2*-item sets), checking support values at each level. The process continues iteratively for k-itemsets until all frequent patterns are discovered.

Building on AIS, the Apriori algorithm [6] introduces a pruning technique that removes candidate itemsets with infrequent subsets before support counting. Both AIS and Apriori require repeated full-database scans, which creates significant time and memory challenges for large datasets [7, 8]. Various enhancements to Apriori have been proposed, including reduced database traversals [9–12], improved pruning strategies [13–15], and techniques for simultaneously mining frequent itemsets [16, 17].

Another prominent method is the FP-Growth algorithm [18], which constructs a compact FP-tree to represent frequent patterns and avoids candidate generation altogether. It requires only a single database scan but introduces its own scalability issues, particularly in memory usage and parallelization. Redundant nodes and sparse entries can further degrade its efficiency when applied to dense datasets.

AIS, Apriori, and FP-Growth represent the most widely studied classical algorithms for MBA. Despite substantial progress in algorithmic optimizations, parallelism, and hardware performance, classical approaches still face fundamental limitations regarding processing time, scalability, and memory consumption [19, 20]. In this context, quantum computing presents a promising paradigm, especially when dealing with big data challenges, by utilising the exponential expansion of the Hilbert space and enabling parallel evaluation of enormous solution spaces [21, 22].

Several quantum approaches have been explored for MBA. One approach uses Quantum Phase Estimation (QPE) [23] to encode transaction data in a superposition state, requiring $2nm + 1$ qubits for n items and m transactions [24, 25]. Another method, proposed by Alasow et al. [26], builds upon Grover's Search algorithm [27]

using quantum counters [28] and comparators [29] applied to Dicke states [30]. This approach demonstrated a reduction in required qubits (e.g., 19 qubits instead of 81) while maintaining performance for small datasets.

Quantum solutions in this domain remain limited in terms of scalability. Notably, existing study [26] rely on generic quantum comparators, which cause the required number of qubits to grow exponentially with the number of encoded elements—leading to serious scalability challenges in practical implementations. To address these limitations, we propose a novel method based on qudit systems. By extending Grover's Search algorithm, Dicke states and quantum comparator into d-level system, proposed method reduces quantum resource requirements while retaining algorithmic fidelity. Our proposed method also further reduces the required quantum resources by incorporating an adaptive comparator specific to qudit systems instead of a generic comparator for efficient itemset evaluation.

This paper consists of 6 sections. Section-1 presents classical and quantum algorithms proposed for MBA to date. Section-2 covers an overview of quantum computation with qudit systems. Also, this section provides details of adaptive quantum comparator on qudit systems. Section-3 presents the proposed quantum method of market basket analysis on qudit systems. Section-4 provides an experiment, while Section-5 includes our findings. The final section is reserved to evaluate the practical importance of the findings obtained.

2 Overview

This section provides insights about qudit systems (aligned with the motivation of this work) due to the limited literature compared to qubits, followed by the Grover Search Algorithm analysis on such systems.

Qudit systems are occasionally described in the literature as high-dimensional computation. In this paper, the terms "qudit systems" and "qudits" will be used interchangeably. When a dimension necessitates specification, we will use the term "d-level" or "d-level system".

2.1 Qudits

To assist readers unfamiliar with quantum computing on qudit systems, it is important to note that while qubits encode binary states $|0\rangle$ and $|1\rangle$, quantum digits – qudits operate on d-level system, where each qudit can exist in a superposition of d-orthonormal basis states:

$$_d|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{d-1}|d-1\rangle, \tag{1}$$

where $\alpha_i \in \mathbb{C}$ and $\sum_i^{d-1} |\alpha_i|^2 = 1$, as required by the normalization condition in quantum mechanics. One can then perform quantum computation using a series of operations that act on a valid quantum state. Such operations are more commonly known as quantum gates. Quantum gates applicable to a qudit can be computed using

the following general unitary transformations.

$$ {}_d\mathrm{U}_{pq} = \sum_{v=0}^{d-1} e^{i\frac{2\pi}{d}vp} |(v+q) \bmod d\rangle\langle v|, \quad p,q \in [0,\ d-1] \tag{2} $$

The transformations given with Equation 2, also known as Weyl operators [31]. These enable the computation of qudit system equivalents of quantum gates on typically used in qubits. With using Equation 2, the following matrices for the X(NOT) gate and the Phase gate on 3-level system:

$$ {}_3\mathrm{U}_{01} = X = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad {}_3\mathrm{U}_{02} = X^2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \tag{3} $$

$$ {}_3\mathrm{U}_{10} = P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{pmatrix}, \qquad {}_3\mathrm{U}_{20} = P^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega^2 & 0 \\ 0 & 0 & \omega^4 \end{pmatrix}, \tag{4} $$

where $\omega = e^{i\frac{2\pi}{d}}$, can be calculated as an illustration. In addition to those gates, the effect of the Hadamard gate, which creates the equal amplitude superposition state commonly utilized in quantum computing, on a single qudit can be observed as follows:

$$ {}_d\mathrm{H}|j\rangle = \frac{1}{\sqrt{d}} \sum_{v=0}^{d-1} \omega^{jv} |v\rangle. \tag{5} $$

2.2 Quantum Resource

A qubits, or qudits, is an abstract concept that enables the development of algorithms irrespective of the physical architecture of quantum computing hardware. The difference between qubits and qudits lies in the number of discrete d-level bases accessible for analysis of the underlying hardware. In a photonic quantum computer, a photon with degrees of freedom $d = 2$ has polarisation restricted to $\{|V\rangle, |H\rangle$, corresponding to the computational bases $|0\rangle, |1\rangle$. In a qudit system, a single photon can be associated to the d-level computational basis $|0\rangle, |1\rangle, \ldots, |d-1\rangle$. In this paper, the term "quantum resource" is used as the minimum number of particles required for quantum algorithms or methods to operate at a selected d-level system. Some studies examined such methodologies to enable comparison between systems of different dimensions or levels [32–34].

2.3 Encoding on Qudits

One of the advantages of the qudits systems is to encode multiple d-ary value within a single quantum resource, allowing a single qudit to store $log_2 d$ bits of classical information. For example in base encoding:

- A qubit (d=2) encodes 1 bit,

- A qutrit (d=3) encodes approximately 1.58 bits,
- An 8-level qudit (d=8) encodes 3 bits.

Thus, using n qudits each with d-level, the total Hilbert space dimension becomes $\mathcal{H}_{\text{qudit}} = d^n$. In comparison, for n qubits, this space is limited to: $\mathcal{H}_{\text{qubit}} = 2^n$. This exponential expansion of the state space means that more complex data structures (e.g., itemsets in MBA) can be represented using fewer physical quantum elements. This is particularly advantageous for applications like market basket analysis, where representing all item combinations or large transaction sets becomes computationally infeasible in traditional qubit system encodings. Readers are encouraged to look at the research conducted by Wack et al., which provides comprehensive analysis of coding in qudit systems [35].

Moreover, qudit systems can potentially reduce circuit depth and quantum gate count, since operations that require multiple gates in qubit systems (e.g., multi-valued logic, multi-level comparisons) can be encoded more efficiently using a smaller number of qudit-based operations. In our implementation, we utilize 7 qudits with *8*-level, resulting in a Hilbert space of size $8^7 = 2,097,152$ bits for encoding. To achieve the same resolution in a qubit systems, at least 21 qubits would be needed, since $2^{21} = 2,097,152$. This illustrates the advantage of making the qudit systems approach more practical and scalable for larger datasets.

2.4 Theoretical Analysis of Grover on Qudits

Grover's Search Algorithm, originally formulated for binary qubit systems, can be extended to operate within a qudit systems to exploit the capabilities of quantum computing. In such systems, the number of possible basis states increases exponentially with both the d-level and the number of qudits n, resulting in a total search space size of d^n.

The core principle of Grover's algorithm is amplitude amplification, where the probability amplitude of target (marked) states is increased iteratively. In the traditional qubit version, the success probability after R iterations is expressed as:

$$P = sin^2((2r+1)\theta), \tag{6}$$

where $\theta = arcsin(\sqrt{(M/N)})$, with M being the number of solutions, and $N = 2^n$ the search space size. In a d-level system, this is generalized to $N = d^n$. When considering a single marked item (M = 1), the angle becomes:

$$\theta = \arcsin(1/\sqrt{d^n}). \tag{7}$$

Consequently, the optimal number of iterations R to maximize the success probability is approximately:

$$R \approx \frac{\pi}{4} \times \sqrt{d^n} \tag{8}$$

This analysis reveals a key advantage: for fixed n, increasing the d-level results in a larger state space, allowing broader encoding capacity without increasing the number of physical quantum elements.

Readers are encouraged to examine the research [36] on the quantum circuit of the Grover Search Algorithm on qudit systems, including its implementation, examples, operator matrix transformations, and more comprehensive theoretical analysis.

2.5 Adaptive Comparator

The quantum circuit model exhibits a structure identical to the fundamental components seen in digital electronics. For instance, fundamental operations like AND, NOT, and OR are represented by X and CNOT gates in a circuit to express binary digital logical operations. Likewise, complex structures such as MULTIPLIER, SWAP, COUNTER, and COMPARE are also present in the circuit model. Indeed, quantum computing encompasses far more than mere digital arithmetic processes; nonetheless, quantum algorithms include them to some extent. This section explains the mapping of the binary notion that executes these operations in the proposed adaptive quantum comparator within qudit systems, as opposed to qubit.

The previous study utilizes generalized quantum comparator circuit for comparing the counter value to minimum support value [26]. The generalized quantum comparator requires a total of $3s + 1$ qubits to compare an s-bit value, including s qubits for comparisons, s ancilla qubits, and 1 output qubit. However, in an MBA problem, the minimum support value for comparison is predetermined and constant, resulting in a rather special scenario. In the specialized comparator (or adaptive as we call it), the bits with the same indices are sequentially compared while evaluating the s-bit counter value and the s-bit support value, both represented in d-base. In every comparison, a multi-value controlled ${}_dX$ gate is added to circuit for execution on the output qudit. If the output qudit's eigenvalue is non-zero, it indicates that the counter exceeds the support value.

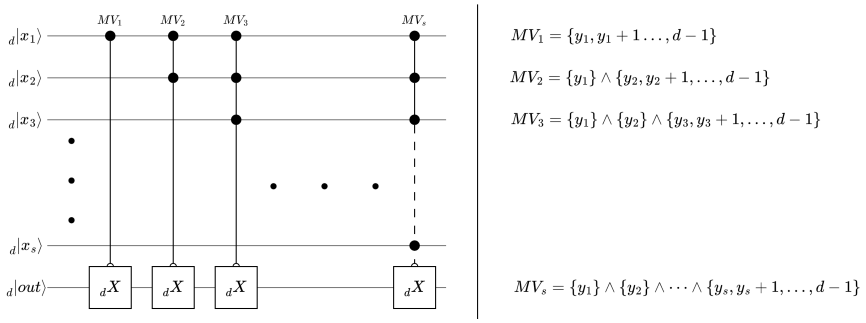


Fig. 1: Adaptive comparator quantum circuit that evaluate s-digit values of $x > y$ in d-level system.

Figure 1 illustrates the $x > y$ adaptive comparison on the d-level quantum circuit. Here, $x = \{x_1, x_2, \ldots, x_s\}$ and $y = \{y_1, y_2, \ldots, y_s\}$ represent s-digit values. In the initial step, a controlled gate ${}_dX$ with control values set to MV_1 is added between the x_1 qudit and the *out* qudit to evaluate the condition $x_1 > y_1$. The MV_1 value set consists

of $\{y_1, y_1+1, \ldots, d-1\}$, with the condition that x_1 is greater than or equal to y_1. If x_1 exceeds y_1, then $x > y$ will already be established. However, the case in which x_1 equals y_1 requires evaluation of $x_2 > y_2$. In the second step, a controlled gate ${}_dX$ with control values set to MV_2 is added between the x_2 qudit and the *out* qudit to evaluate the condition $x_2 > y_2$. However, since this step also requires the $x_1 = y_1$ condition to be met, the MV_2 value set must be $\{y_1\}$ for the x_1 qudit and $\{y_2, y_1+2, ...d-1\}$ for the x_2 qudit. Within s steps, the quantum evaluation of $x > y$ is performed by adding s multi-value controlled ${}_dX$ gates to the quantum circuit. If any of the controlled ${}_dX$ gates satisfy the condition, the inequality $x > y$ will be verified. When all conditions are met, it will be known that $x = y + 1$. In this case, the *out* qudit will be in the $|s\rangle$ state. Therefore, the chosen d-level system must satisfy $d > s$ to ensure that the state of the *out* qudit does not revert back to $|0\rangle$.

3 Qudit Approach on MBA

This study generalizes the Grover Search algorithm, a component of the previous approach [26], to d-level and replaces the general quantum comparator circuit with a adaptive comparator to further reduce the number of qudits. To proof of out proposed method, let $G(T, I)$ represent a database as shown in Table 1, comprising the transaction set T, which is a subset of the itemset I.

Table 1: $G(T, I)$ Database, consist of n transactions and m items.

Transactions	Items
T_1	T_0 Items $\subseteq \{I_1, I_2, I_3, \ldots, I_m\}$
T_2	T_1 Items $\subseteq \{I_1, I_2, I_3, \ldots, I_m\}$
T_3	T_2 Items $\subseteq \{I_1, I_2, I_3, \ldots, I_m\}$
$\vdots$	$\vdots$
T_n	T_n Items $\subseteq \{I_1, I_2, I_3, \ldots, I_m\}$

Initially, a truth table for the items associated with each item must be constructed, as demonstrated in Table 2.

Table 2: Truth table of transactions and items for $G(T, I)$ dataset. $(TI_{ij} \in \{0, 1\})$

	I_1	I_2	I_3	$\ldots$	I_m
T_1	TI_{11}	TI_{12}	TI_{13}	$\ldots$	TI_{1m}
T_2	TI_{21}	TI_{22}	TI_{23}	$\ldots$	TI_{2m}
T_3	TI_{31}	TI_{32}	TI_{33}	$\ldots$	TI_{3m}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
T_n	TI_{n1}	TI_{n2}	TI_{n3}	$\ldots$	TI_{nm}

Table 2 checks the existence of corresponding items within each transaction. If a transaction T_i includes item I_j, then TI_{ij} equals *1*; otherwise TI_{ij} equals *0*. These transactions will be encoded into the quantum circuit using Dicke states. Thus, all transactions must contain minimum support items at maximum. Algorithm 1 presents the pseudo-code for the procedure that generates new transactions containing a minimum support of items, rather than transactions with a surplus of minimum support items. With the algorithm, a new database $G'(T', I)$ can be generated.

Algorithm 1 sequentially checks all transactions for the number of items. Transactions with an item count below the minimum support(ms) value are eliminated. When a transaction's item count exceeds the ms threshold, new subsets of the transaction's elements are generated, each containing the ms quantity of items. Thus, there will be no transaction in the database that contains more or less than the ms value.

Algorithm 1 Transform Transactions Based on Minimum Support Value

```
Set ms ← minimum support
Initialize G ← [T_0, T_1, T_2, ..., T_t]
Initialize G' as an empty database

for each transaction in G do
    if length of transaction ≤ ms then
        Append transaction to G'
    else
        Initialize new transaction T' as an empty item list
        Initialize indices ← list of integers from 0 to ms - 1
        while True do
            Set current_combination ← items in transaction at positions specified by indices
            Append current_combination to T'
            for i from ms - 1 to 0 (inclusive) do
                if indices[i] ≠ i + length of transaction - ms then
                    break the for loop
                end if
            end for
            if the for loop did not break then then
                break the while loop
            end if
            Increment indices[i] by 1
            for j from i + 1 to ms - 1 (inclusive) do
                Set indices[j] ← indices[j − 1] + 1
            end for
        end while
        Append T' to G'
    end if
end for
```

The new number of transactions n' in the new $G'(T', I)$ database, generated from Algorithm 1, can be calculated as follows.

$$n' = \sum_{v=1}^{n} f(\text{length } T_v) \tag{9}$$

$$f(x) = \begin{cases} 0 & \text{if } x < \text{ms} \\ 1 & \text{if } x = \text{ms} \\ \binom{x}{\text{ms}} & \text{if } x > \text{ms} \end{cases} \tag{10}$$

Upon the revision of Table 2 with the new database, $|D_{n'}^{ms}\rangle$ Dicke state;

$$\begin{aligned} |D_m^{ms}\rangle \; = |T'I_{11}\; T'I_{12}\; \ldots\; T'I_{1m}\rangle \; + \; |T'I_{21}\; T'I_{22}\; \ldots\; T'I_{2m}\rangle \; + \\ |T'I_{31}\; T'I_{32}\; \ldots\; T'I_{3m}\rangle \; + \; |T'I_{41}\; T'I_{42}\; \ldots\; T'I_{4m}\rangle \; + \\ \vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ |T'I_{n'1}\; T'I_{n'2}\; \ldots\; T'I_{n'm}\rangle, \end{aligned} \tag{11}$$

can be calculated. The quantum circuit begins by initializing this $|D_s^{ms}\rangle$ quantum state in solving market basket analysis using quantum computing. Then, the quantum circuit proceeds with the execution of the Grover Search algorithm.

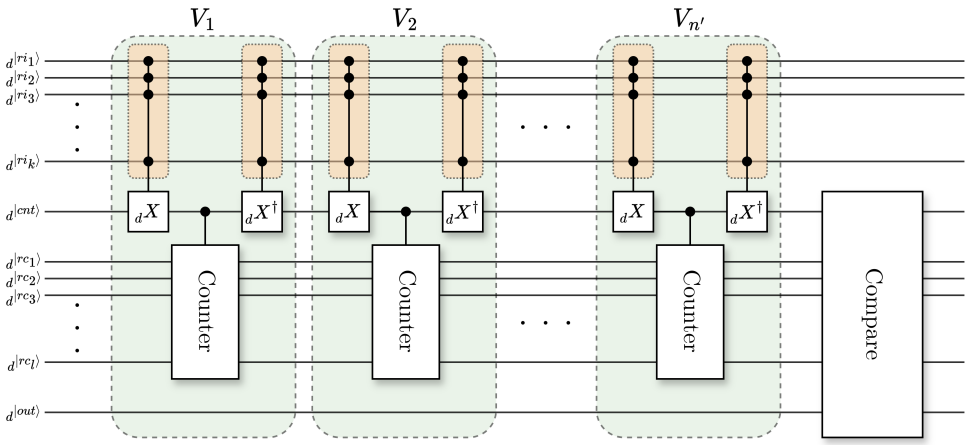


Fig. 2: Oracle Circuit for MBA implementation on qudit systems. The $k+l+2$ qudit circuit includes an n' counter and a compare block.

Figure 2 illustrates the Oracle circuit of the Grover Algorithm necessary for MBA implementation. The circuit has total $k + l + 2$ qudits: $k = \lceil \log_d 2^m \rceil$ qudits for the registers denoted by $_d|ri\rangle$, $l = \lceil \log_d n' \rceil$ qudits allocated for the counter denoted by $_d|rc\rangle$, one $_d|out\rangle$ qudit for comparing counters, and one $_d|cnt\rangle$ control qudit. The circuit

possesses n' blocks that trigger the counter based on the control values V_i of the register qudits. Here, the multi-value control set V_i equals $V_i = \{T'I_{i1},\ T'I_{i2},\ \ldots,\ T'I_{im}\}$ in d-base. The circuit concludes with an adaptive quantum comparator previously described in Section 2.5.

4 Experiment

In order to illustrate the quantum solution of the MBA problem, a database that consists of 9 items and 64 transactions was constructed by random sampling. A section of the database is given in Table 3 below. The database is imported similarly to Table 2. For simplicity, all 64 transactions in this experiment are designed to contain exactly 5 items. The transaction count was set at 64 in 8-level to eliminate padding, while the item count was set at 9 to ensure to perform simulations on the classical computer. Also, since 8 is an integer power of 2, calculating the matrix of gates on 8-level system is easier.

Table 3: A head and tail section of the database used in the experiment.

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
T_0	1	0	1	0	0	1	0	1	1
T_1	1	0	1	1	0	1	0	0	1
T_2	0	0	1	0	1	1	1	0	1
T_3	1	0	1	0	1	0	1	0	1
T_4	1	1	0	1	0	1	0	1	0
...	...	...	...	...	...	...	...	...	...
T_{59}	0	1	0	1	0	1	1	1	0
T_{60}	1	1	0	0	1	1	0	1	0
T_{61}	0	0	1	1	0	1	1	0	1
T_{62}	0	0	1	0	1	1	1	0	1
T_{63}	1	0	1	0	1	0	1	0	1

If the database transactions contain varying quantities of items, then Algorithm 1 must be executed. Algorithm 1 guarantees that all transactions contain no more than ms items. However, it derives new transactions from the database.

Figure 3 presents the quantum circuit designed to simulate the experiment. Figure 3a illustrates that this application requires 2 register qudit denoted as ${}_8|r\rangle$, a counter controller qudit denoted as ${}_8|cnd\rangle$, 2 counter qudit denoted as ${}_d|c\rangle$, and a ${}_8|out\rangle$ qudit. The quantum circuit of the stated experiment start with encoding the previously obtained ${}_8|D_5^9\rangle$ state into a 3-register qudits in 8-level system. Then, the Oracle operator designated for the experiment database is integrated into the circuit. Grover's Diffusion component is applied solely when the ${}_8|out\rangle$ qudit is not equal to the $|0\rangle$ state. The Oracle operator must be reiterated at least 4 times in conjunction with Diffusion. The minimum number of iterations can be determined using the $R \leq \lceil \frac{\pi}{4}\sqrt{N/M} \rceil$ equation. In the experiment, the quantity of Dicke states is prepared to match the number of transactions n', $N = 64$. In the classical solution, the collection of items surpassing the minimum support threshold is $M = 3$. At the final stage of the circuit, the register qudits are measured.

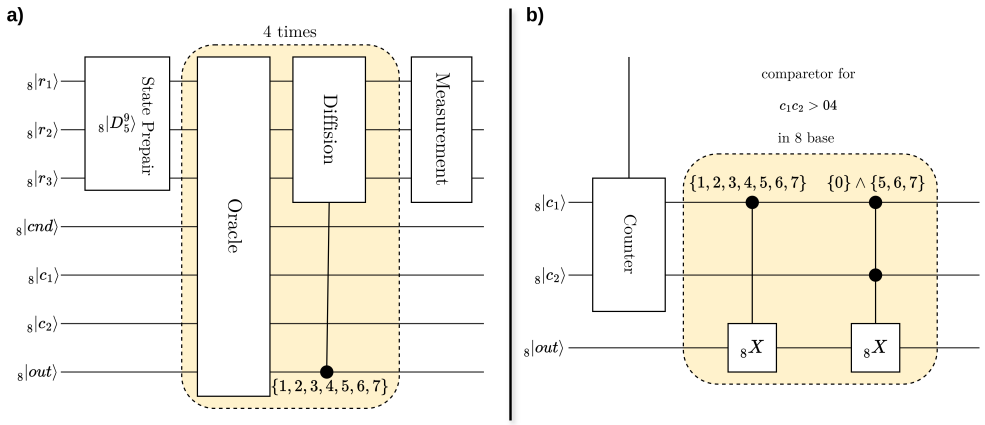


Fig. 3: Implementation of proposed method for 9 item, 64 transaction database in *8*-level. **a-)** Full circuit of implementation using 7 qudits. **b-)** Adaptive comparator circuit for ms = 5 in the Oracle part.

Figure 3b illustrates the adaptive comparator with a minimum support value of 5 at the end of the Oracle operator. The figure illustrates that if c_1 digit exceeds 1, the counter will beyond the minimum support threshold. However, if the value of c_1 digit is 0, then the condition that c_2 exceeds 4 $(ms-1)$ must be checked. The comparator circuit is generated by a selected value for *ms* as seen in the figure.

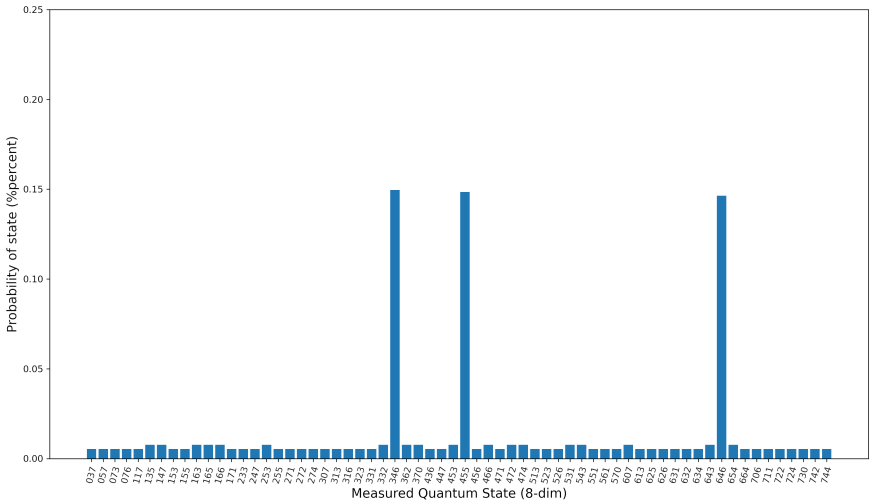


Fig. 4: Simulation result of the experiment on Cirq[37]. Results indicate the most likely measurement outcomes are $_8|346\rangle$, $_8|455\rangle$ and $_8|646\rangle$. Probabilities lower than 0.0005 are not included.

Post-Processing of Results. Grover's algorithm leans more deterministic in terms of the probabilistic distribution of measurement results. While the sought elements obtain peak probability values in the results, the unsought– unwanted elements have very low probabilistic values. Despite the quantum circuit in this study starting with the Dicke state rather than an equal superposition, the Grover algorithm still gives results with a similar trend. While the sought elements denote the most commonly occurring elements in this study, it is essential to interpret the numeric values of measured states. To achieve this, it is necessary to first convert the highest probability measurement values obtained to base-*2* strings. The length of the binary array from the conversion will be equal to the number of elements. Similarly, the number of bits with a value of "1" in this bit array will equal *ms*. The indices of these bits with a value of "1" will be the same as the indices of the ordered elements in the database.

The experiment circuit simulation result is given in Figure 4. The figure shows that the highest values of measurement probabilities of 0.146, 0.149 and 0.148 were obtained for ${}_8|346\rangle$, ${}_8|455\rangle$ and ${}_8|646\rangle$, respectively. According to post-processing, the obtained measurement results are first converted to the base-*2* string equivalent values of $346_8 = 011100110_2$ $455_8 = 100101101_2$ and $646_8 = 110100110_2$. Then, when the indices of "1" in these bit arrays are mapped to the same index as the elements, the following 5-element element lists are obtained: $\{I1, I2, I3, I6, I7\}$, $\{I0, I3, I5, I6, I8\}$ and $\{I0, I1, I3, I6, I7\}$. These sets are the most repeated items on all transactions.

5 Results

Our proposed quantum MBA method has an advantage over previous methods in terms of usage of quantum resource. The comparisons among approaches or methods are given in Table 4.

Table 4: Comparison of the number of quantum resources used in the MBA quantum solution with our proposed method in this study and the previous studies.

Method	Quantum Resources
Previous method [24] (*2*-level)	1152
Previous method [26] (*2*-level)	32
Previous method [26] on qudits (*8*-level)	11
Our proposed method (*8*-level)	7

Also, Table 4 indicates that, if the previous method [26] were extended to qudit systems without any algorithm changes, 11 qudits would be required. Thanks to the adaptive comparator, the same comparison can be done using 1 qudit instead of 5 qudits. The reason for this is that when a qudit has more capacity to hold data than the threshold value, then comparison can be made with only a few controlled gates.

(a) **(b)**

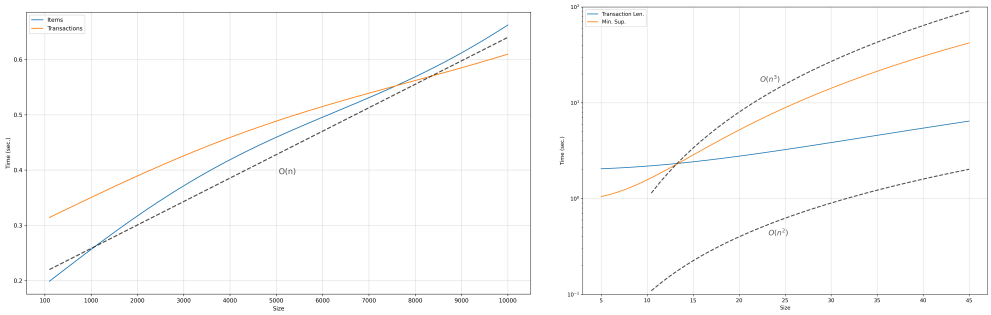


Fig. 5: Comparison of the classical data preparation process with Algorithm 1 in terms of runtime. **(a)** Performance of the algorithm over time (sec.) in terms of number of transactions and number of items. **(b)** Performance of the algorithm over time in terms of maximum transaction length and minimum support threshold.

An additional matter that deserves discussion in the results section is the classical performance of Algorithm 1. Numerous studies in the literature using classical-quantum hybrid components fail to assess the performance of the classical part in their analyses. However, as we consider these analyses crucial for the execution of quantum algorithms, the possibility of the classical data preparation part outlined in Algorithm 1 should be assessed as well. Algorithm 1 comprises four parameters that impact classical performance: the amount of transactions, the quantity of items, the maximum transaction length, and the minimum support threshold. The parameters were examined independently in relation to time performances of the algorithm. Figure 5a illustrates that the amount of transactions and the quantity of items in the database were evaluated up to 10,000. It was noted that at these circumstances, Algorithm 1 was resolved in a maximum of 0.68 seconds. This leads to a convergence to a line characterised by a time complexity of $O(n)$, as depicted in the figure. Correspondingly, the analytical results for the maximum transaction length and minimum support threshold parameters are presented in Figure 5b. This results, presented on a logarithmic scale, indicates that the algorithm's time complexity for these values lies between $O(n^2)$ and $O(n^3)$. In a practical or real world MBA scenario, given that the minimum support value is intended to be minimal (e.g., identifying which three products a customer purchased together) and the transaction lengths are relatively short (e.g., the total number of products a customer bought), we believe that Algorithm 1 can be executed on current computers within in a reasonably short time.

6 Conclusion

While the proposed method in this study has been proven accurate for quantum computation with qudits, unfortunately, it has not been tested on a quantum computer. This is primarily due to the limited analysis capabilities of current quantum computers on such systems. Furthermore, although the quantum gates used in this study are generalized, it remains an open question whether a future quantum computer capable of analysis could utilize these standards on qudits. Indeed, many studies in the current

literature mostly analyse qudit systems analytically, avoiding a one-to-one comparison with traditional quantum computation. Still, this study contributes to this limited literature by demonstrating the potential of quantum computation with qudits as an efficient and resource-optimized approach to address challenges in MBA.

In this study, adaptive comparator circuit used in the proposed method in order to reduce the number of required quantum resources. But this comparator is valid only if the value being assessed remains constant throughout the given problem. When seen in this way, it may be regarded as highly constrained. Nonetheless, there exist extensive classical and quantum algorithms, as well as analytical techniques, that require comparison with a constant value, exemplified by MBA in this study. Consequently, we believe that this comparator can be merged with other algorithms to address similar situations.

The results emphasize that quantum computing with qudits significantly minimizes the exponential growth of quantum resource requirements and improves computational efficiency when dealing with large datasets. Experimental results validated by simulations in the Cirq framework confirm the feasibility and effectiveness of the proposed method. Within its limitations, this work could contribute to the continued advancement of quantum computing in data mining by providing a scalable, resource-efficient solution, leading to broader applications in big data analysis and beyond.

Author Contributions The author Semi Ceylan, conceived the research idea and designed the methodology. The authors Semi Ceylan and Sait Can Yücebaş implemented the experiments and performed data analysis, contributed to the interpretation of results and manuscript preparation. All authors read and approved the final manuscript.

Funding The authors did not receive support from any organization for the submitted work.

Data Availability The dataset (although synthetically generated) and findings obtained in this study are available upon request.

Declarations

Competing interests The authors declare no competing interests.

References

[1] Chen, Y.-L., Tang, K., Shen, R.-J., Hu, Y.-H.: Market basket analysis in a multiple store environment. Decision support systems **40**(2), 339–354 (2005)

[2] Sherly, K., Nedunchezhian, R.: A improved incremental and interactive frequent pattern mining techniques for market basket analysis and fraud detection in distributed and parallel systems. Indian Journal of Science and Technology **8**(18), 1–12 (2015)

[3] Rao, A.B., Kiran, J.S.: Application of market–basket analysis on healthcare. International Journal of System Assurance Engineering and Management **14**(Suppl 4), 924–929 (2023)

[4] Patwary, A.H., Eshan, M.T., Debnath, P., Sattar, A.: Market basket analysis approach to machine learning. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–9 (2021). IEEE

[5] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)

[6] Agrawal, R.: Fast algorithms for mining association rules. (1994). VLDB

[7] Purdom, P.W., Van Gucht, D., Groth, D.P.: Average-case performance of the apriori algorithm. SIAM Journal on Computing **33**(5), 1223–1260 (2004)

[8] Xie, H., *et al.*: Research and case analysis of apriori algorithm based on mining frequent item-sets. Open Journal of Social Sciences **9**(04), 458 (2021)

[9] Arcos, J.R.D., Hernandez, A.A.: Efficient apriori algorithm using enhanced transaction reduction approach. In: 2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA), pp. 97–101 (2019). IEEE

[10] Rao, S., Gupta, P.: Implementing improved algorithm over apriori data mining association rule algorithm. Ijcst **3**(1), 489–493 (2012)

[11] AL-Zawaidah, F.H., Jbara, Y.H., Marwan, A.: An improved algorithm for mining association rules in large databases. World of Computer science and information technology journal **1**(7), 311–316 (2011)

[12] Singh, J., Ram, H., Sodhi, D.: Improving efficiency of apriori algorithm using transaction reduction. International Journal of Scientific and Research Publications **3**(1), 1–4 (2013)

[13] Borgelt, C.: Recursion pruning for the apriori algorithm. In: FIMI, vol. 126 (2004)

[14] Nagpal, S.: Improved apriori algorithm using logarithmic decoding and pruning. International Journal of Engineering Research and Applications **2**(3), 2569–2572 (2012)

[15] Ji, L., Zhang, B., Li, J.: A new improvement on apriori algorithm. In: 2006 International Conference on Computational Intelligence and Security, vol. 1, pp. 840–844 (2006). IEEE

[16] Chen, Z., Cai, S., Song, Q., Zhu, C.: An improved apriori algorithm based on

pruning optimization and transaction reduction. In: 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), pp. 1908–1911 (2011). IEEE

[17] Yuan, X.: An improved apriori algorithm for mining association rules. In: AIP Conference Proceedings, vol. 1820 (2017). AIP Publishing

[18] Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. ACM sigmod record **29**(2), 1–12 (2000)

[19] Sjarif, N.N.A., Azmi, N.F.M., Yuhaniz, S.S., Wong, D.H.-T.: A review of market basket analysis on business intelligence and data mining. International journal of business intelligence and data mining **18**(3), 383–394 (2021)

[20] Sharma, A., Ganpati, A.: Association rule mining algorithms: a comparative review. Int. Res. J. Eng. Technol **8**(11), 848–853 (2021)

[21] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G., Buell, D.A., *et al.*: Quantum supremacy using a programmable superconducting processor. Nature **574**(7779), 505–510 (2019)

[22] Villalonga, B., Lyakh, D., Boixo, S., Neven, H., Humble, T.S., Biswas, R., Rieffel, E.G., Ho, A., Mandrà, S.: Establishing the quantum supremacy frontier with a 281 pflop/s simulation. Quantum Science and Technology **5**(3), 034003 (2020)

[23] Kitaev, A.Y., Shen, A., Vyalyi, M.N.: Classical and quantum computation (2002)

[24] Yu, C.-H., Gao, F., Wang, Q.-L., Wen, Q.-Y.: Quantum algorithm for association rules mining. Physical review A **94**(4) (2016)

[25] Yu, C.-H.: Experimental implementation of quantum algorithm for association rules mining. IEEE Journal on Emerging and Selected Topics in Circuits and Systems **12**(3), 676–684 (2022)

[26] Alasow, A., Perkowski, M.: Quantum algorithm for mining frequent patterns for association rule mining. Journal of Quantum Information Science (2023)

[27] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)

[28] Vedral, V., Barenco, A., Ekert, A.: Quantum networks for elementary arithmetic operations. Physical Review A **54**(1), 147 (1996)

[29] Al-Rabadi, A.N.: Closed-system quantum logic network implementation of the viterbi algorithm. Facta universitatis-series: Electronics and Energetics **22**(1), 1–33 (2009)

[30] Nepomechie, R.I.: Qudit dicke state preparation. Quantum Information and Computation **24**(1&2), 0037–0056 (2024)

[31] Patera, J., Zassenhaus, H.: The pauli matrices in n dimensions and finest gradings of simple lie algebras of type a n- 1. Journal of Mathematical Physics **29**(3), 665–673 (1988)

[32] Janković, D., Hartmann, J.-G., Ruben, M., Hervieux, P.-A.: Noisy qudit vs multiple qubits: conditions on gate efficiency for enhancing fidelity. npj Quantum Information **10**(1), 59 (2024)

[33] Esposito, C., Di Colandrea, F., Hoch, F., Carvacho, G., Cardano, F., Spagnolo, N., Marrucci, L., Sciarrino, F.: Generation of high-dimensional qudit quantum states via two-dimensional quantum walks. Physical Review Research **5**(4), 043025 (2023)

[34] Youvan, D.: Qudits over qubits: The case for higher dimensional quantum computing (2023) https://doi.org/10.13140/RG.2.2.24460.90245

[35] Wach, N.L., Rudolph, M.S., Jendrzejewski, F., Schmitt, S.: Data re-uploading with a single qudit. Quantum Machine Intelligence **5**(2), 36 (2023)

[36] Acar, E., Gündüz, S., Akpınar, G., Yılmaz, İ.: High-dimensional grover multi-target search algorithm on cirq. The European Physical Journal Plus **137**(2), 1–9 (2022)

[37] Developers, C.: Cirq. https://doi.org/10.5281/zenodo.4750446 . https://doi.org/10.5281/zenodo.4750446