Wiley
Wireless Communications and Mobile Computing
Volume 2024, Article ID 7716976, 14 pages
https://doi.org/10.1155/2024/7716976

WILEY

Research Article

Improvement of Apriori Algorithm Using Parallelization Technique on Multi-CPU and GPU Topology

Hooman Bavarsad Salehpour,[1] Hamid Haj Seyyed Javadi,[2] Parvaneh Asghari,[3] and Mohammad Ebrahim Shiri Ahmad Abadi[4]

[1]*Department of Computer Engineering, Borujerd Branch, Islamic Azad University, Borujerd, Iran*
[2]*Department of Mathematics and Computer Science, Shahed University, Tehran, Iran*
[3]*Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran*
[4]*Department of Mathematics and Computer Science, Amirkabir University, Tehran, Iran*

Correspondence should be addressed to Parvaneh Asghari; p_asghari@iauctb.ac.ir

Received 26 April 2023; Revised 26 October 2023; Accepted 9 March 2024; Published 31 May 2024

Academic Editor: L. J. García Villalba

In the domain of data mining, the extraction of frequent patterns from expansive datasets remains a daunting task, compounded by the intricacies of temporal and spatial dimensions. While the Apriori algorithm is seminal in this area, its constraints are accentuated when navigating larger datasets. In response, we introduce an avant-garde solution that leverages parallel network topologies and GPUs. At the heart of our method are two salient features: (1) the use of parallel processing to expedite the realization of optimal results and (2) the integration of the cat and mouse-based optimizer (CMBO) algorithm, an astute algorithm mirroring the instinctual dynamics between predatory cats and evasive mice. This optimizer is structured around a biphasic model: an initial aggressive pursuit by the cats and a subsequent calculated evasion by the mice. This structure is enriched by classifying agents using their objective function scores. Complementing this, our architectural blueprint seamlessly amalgamates dual Nvidia graphics cards in a parallel configuration, establishing a marked ascendancy over conventional CPUs. In amalgamation, our approach not only rectifies the inherent shortfalls of the Apriori algorithm but also accentuates the extraction of association rules, pinpointing frequent patterns with enhanced precision. A comprehensive evaluation across a spectrum of network topologies explains their respective merits and demerits. Set against the benchmark of the Apriori algorithm, our method conspicuously outperforms in terms of speed and effectiveness, heralding a significant stride forward in data mining research.

1. Introduction

Data mining is a valuable tool for detecting hidden patterns and uncovering relationships between data [1]. For small databases with limited data volume, simple queries embedded in different databases can effectively extract useful information and uncover certain data rules [2]. However, discovering and reporting hidden patterns in big data requires significant resources and expertise in data exploration and analysis [3]. By definition, data mining converts low-level knowledge into high-level, comprehensible knowledge and often reveals previously unknown and beneficial patterns [4]. Data mining encompasses various techniques and methods for discovering knowledge and pattern recognition, such as discovering association rules [5], clustering [6], classification [7], time series [8], artificial neural networks [9], machine learning [10], and decision trees [11].

Various algorithms have been developed for extracting association rules in data mining. A foundational algorithm for association rule mining (ARM) is introduced in [12]. This method extracts only one item from the rules using an algorithm based on the discovery of simple rules. The algorithm employs iterative searches of the data to determine the support for each possible candidate. In [13], the search operation is conducted in two steps. First, the most frequent item sets are identified, and in the second step, the rules in the frequent subsets are extracted using iterative procedures, one rule at a time [14] introduced an algorithm based on direct hashing and pruning (DHP). This method produces

large item sets using the HASH technique to represent transactions, leading to a reduction in the database's size. The use of the clustering technique to avoid duplicate search mechanisms in the database is presented in [15]. The algorithm introduced in [16] involves searching the database twice to achieve the necessary support. Another algorithm, introduced in [17], is based on the node-set data structure and preorder coding (POC) tree to discover frequent data sets. Additionally [18], proposes an algorithm for exploring closed, frequent sequences of item sets. This algorithm employs sparse and vertical id-list data structures to represent the dataset, along with a new one-step methodology for controlling sequence closure and search space pruning.

The Apriori algorithm is one of the fundamental algorithms for detecting association rules. It calculates and extracts all k-member subsets of a k-product set [19]. This algorithm identifies frequent single-member products after a complete search of the database. Then, using these frequent single-member products, it identifies two-member sets and turns these sets into candidates for counting. In the next step, the database is searched to count the number of occurrences of these sets. The primary challenges in algorithms like Apriori are the quality of the output, the extracted relations, and the algorithm's running time. Increasing the size of items or products can significantly increase the traversal time of algorithms like Apriori. As a result, parallel processing methods can considerably reduce the running time of extracting frequent patterns. Another method for parallelizing and speeding up calculations is using the parallel thread library and running the program on the central processing unit (CPU). However, parallelizing algorithms like Apriori based on the thread library may not achieve a high level of parallelism due to CPU sharing between different processes. Moreover, the thread library must often wait in a queue to access CPU cores [20]. A recent trend involves using a graphics processor unit (GPU) instead of the primary processor to parallelize and accelerate algorithms.

Fang et al. [21] presented two distinct implementations of the Apriori method for extracting association rules on next-generation GPUs. Their method leverages SIMD (single instruction, multiple data) architectures in GPUs. Zhang et al. [6] introduced GPApriori, wherein a GPU is optimized for association rule extraction computations. Their implementation, executed on an Nvidia Tesla T10 graphic processor, achieves up to a 100x speedup. In [22], Silvestri and Orlando proposed two GPU parallelization strategies and performed extensive experiments using real-world data to assess this method's efficacy. Fournier-Viger et al. [23] introduced a rapid association rule extraction method for big data called GMiner. However, Apriori has a disadvantage related to its high computational time, as it requires repeated database scans to search for each combination of frequent item sets. In the second proposed method of this study, a new hybrid algorithm based on Apriori and cat-and-mouse optimization aims to enhance performance further. Simulation outcomes suggest that combining the cat-and-mouse and Apriori-based optimization algorithms yields more detailed information, faster calculation speeds, and reduced energy consumption compared to the standard Apriori algorithm.

To enhance the accuracy of the classifier produced by Ant-Miner, the authors in [24] combined ACO with the PSO algorithm. Additionally [25], introduced a new sequential coverage strategy for Ant-Miner to mitigate the rule interaction problem. To counteract premature convergence to local optima in ACO, an ACO-based classification algorithm named Ant-MinerPAE, grounded on pheromone absorption and removal, was proposed in [26]. Concurrently, Ant-Miner faces a deficiency in exploitation due to the absence of a local search [27]. To tackle this, ILS-AntMiner was recently introduced [27]. Djenouri et al. [28] proposed a distributed method based on evolutionary fuzzy systems to extract and consolidate emerging descriptor patterns in data streams from various sources. Initially, an evolutionary algorithm for efficient data processing is introduced to extract emerging patterns from the data streams produced by each device, culminating in a local model for each stream. Subsequently, multiple fusion methods are presented to amalgamate these patterns and formulate the global model. A comprehensive experimental study was undertaken to assess this evolutionary algorithm's efficacy in extracting high-caliber emergent patterns and its aptitude for addressing conceptual drift. The quality of the proposed fusion methods was also scrutinized.

In this context, the moth-flame optimization algorithm is renowned for addressing optimization problems across diverse fields, credited to its straightforward structure and effortless implementation. However, MFO often struggles to strike a balance between the exploration and exploitation processes and grapples with a lack of population diversity during the exploration phase, especially when addressing intricate engineering optimization challenges. To surmount these challenges, Zhao et al. [29] introduced a multiclass improved moth-flame optimization (MIMFO) algorithm. In MIMFO, the population undergoes restructuring through a chaotic grouping mechanism and a dynamic regrouping approach, enhancing the grouping quality and diversifying the population. Spiral search and linear search are executed for two subgroups to bolster search efficacy and harmonize exploration and exploitation. Additionally, a Gaussian mutation is employed to generate the flame, which hastens convergence and augments exploration prowess. The efficacy of MIMFO was tested on 13 benchmark problems of varying dimensions and CEC 2014 test performances, with the outcomes indicating MIMFO's pronounced superiority over other swarm intelligence algorithms and MFO variants in terms of optimal performance and global convergence. MIMFO's prowess in addressing 57 engineering-constrained optimization challenges further underscores its capability in effectively tackling real-world engineering issues.

The pigeon-inspired optimization (PIO) algorithm, a paradigm of intelligent optimization algorithms, draws inspiration from the navigational behavior exhibited by pigeon flocks. PIO outshines other algorithms when addressing numerous optimization challenges. Yet, its efficacy wanes when tackling large-scale intricate optimization problems, and its runtime is protracted. The performance of swarm-based optimization algorithms like PIO can be enhanced through parallel processing, and their hardware implementation prerequisites can be streamlined to expedite execution times. Garcia-Vico et al. [30] put forth a

hardware model rooted in FPGA-based PIO. This approach centers on multi-individual and multidimensional parallelism in pigeon populations. To achieve further acceleration, the method incorporates a parallel bubble sort algorithm and a multiply-accumulator (MAC) pipeline design. Simulation outcomes illustrate that the FPGA-based PIO implementation can markedly bolster PIO's computational prowess and adeptly navigate complex practical challenges.

In this paper, we tackle the challenges associated with extracting frequent patterns from large datasets, especially when complicated by temporal and spatial dimensions. We identify the limitations of the established Apriori algorithm, particularly when applied to more extensive datasets, and in response, present an innovative solution that leverages parallel network topologies and GPU capabilities. Two key features define our approach: first, the use of parallel processing to achieve faster results, and second, the incorporation of the novel CMBO algorithm, which mirrors the natural dynamics between predatory cats and evasive mice. This CMBO algorithm is characterized by a unique biphasic model and is further refined by classifying agents based on objective function scores. In addition, our model's architectural design integrates dual Nvidia graphics cards, giving it a clear advantage over traditional CPU-driven methods. Through our efforts, we not only address the limitations of the Apriori algorithm but also enhance the accuracy of association rule extraction. We subjected our model to rigorous testing across various datasets, and the results consistently demonstrate its superior speed and effectiveness against established benchmarks, signaling a notable advancement in data mining methodologies.

The main contributions of the proposed model are:

(1) The model presents an avant-garde approach that caters specifically to the challenges posed by expansive datasets, addressing both their temporal and spatial complexities. The model utilizes parallel processing, which speeds up the computation and realization of optimal results.
(2) We introduce the CMBO algorithm. This algorithm stands out due to its biphasic model that emulates the instinctual dynamics between predatory cats and evasive mice. It is further refined by the unique method of classifying agents based on their objective function scores.
(3) The architectural design of the model integrates dual Nvidia graphics cards in a parallel setup. This configuration offers a significant advantage over traditional CPU-based systems in terms of speed and computational efficiency.

2. The Proposed Method

In this study, a new optimization algorithm called CMBO, which imitates the natural behavior of cats and mice, is presented. In the proposed CMBO, the movements of cats toward mice, as well as the escape of mice to a safe place, are simulated. Mathematical modeling and CMBO formulation are proposed for implementation of optimization problems. Finally, performing CMBO is evaluated on a standard set of three different objective functions, including unimodal, high-dimensional multimodal, and fixed-dimensional multimodal.

2.1. Cat and Mouse Optimizer Algorithm. In this section, after stating the theory of the CMBO algorithm, its mathematical model is presented for various optimization problems. CMBO is a population-based algorithm inspired by the natural behaviors of cats attacking mice and mice escaping to safety. The search agents in the proposed algorithm are divided into two groups: cats and mice. Search agents scan the problem space with their random movements. The proposed algorithm updates the population of individuals in two consecutive steps. In the first and second phases, the movement of cats toward mice and the escape of mice to a safe place are modeled, respectively.

Regarding mathematics, each individual in the population is a representation of a proposed solution to the problem. Each individual in the population determines the values of the problem variables according to their position in the search space. Therefore, each individual in the population is a vector that determines the values of the variables in the problem. The population of the algorithm is determined using a matrix called the population matrix, according to Equation 1:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m}, \tag{1}$$

where X is the CMBO population matrix, X_i is the i^{th} search factor, $x_{i,d}$ is the problem variable value d obtained by the i^{th} search factor, and N and m are the number of population individuals and the number of problem variables, respectively. The initial values for the mouse population matrix were randomly generated within the boundaries of the problem space.

Each individual in the population determines the proposed values for the variables of the problem. Therefore, a value for the objective function is determined for each individual in the population. The obtained values for the objective function are represented using the vector of Equation 2:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1}, \tag{2}$$

where F is the vector of objective function values and F_i is the objective function value for the i^{th} search factor.

Based on the resulting values for the objective functions, the population individuals are ranked from the best individual with the lowest objective function value to the worst population individual with the highest objective function value. The ranked population matrices and the ranked objective functions are obtained using Equations 3 and 4:

$$X = \begin{bmatrix} X_1^S \\ \vdots \\ X_i^S \\ \vdots \\ X_N^S \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1}^S & \cdots & x_{1,d}^S & \cdots & x_{1,m}^S \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1}^S & \cdots & x_{i,d}^S & \cdots & x_{i,m}^S \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,d}^S & \cdots & x_{N,d}^S & \cdots & x_{N,m}^S \end{bmatrix}_{N\times m}, \tag{3}$$

$$F^S = \begin{bmatrix} F_1^S & \min(F) \\ \vdots & \vdots \\ F_N^S & \max(F) \end{bmatrix}_{N\times 1}, \tag{4}$$

where X^S is the ranked population matrix based on the value of the objective function, X_i^S is the tenth individual of the ranked population matrix, $x_{i,d}^s$ is the value of the problem variable d obtained by the i^{th} search factor from the ranked population matrix, and F^S is the ranked matrix.

For the CMBO model, the time complexity is $O(n^2)$ owing to its iterative nature and the underlying processes that examine pairwise relationships between entities in the dataset. This makes it relatively efficient for sizable datasets, as evidenced by our evaluations.

2.2. Objective Function Vector. In CMBO, we assume that half of the population individuals with higher values for the objective function comprise the mouse population, while the other half of the population individuals with lower values for the objective function comprise the cat population. Based on this assumption, the populations of mice and cats are determined according to Equations 5 and 6, respectively.

$$\begin{aligned} M &= \begin{bmatrix} M_1 = X_1^S \\ \vdots \\ M_i = X_i^S \\ \vdots \\ M_{Nm} = X_{N_m}^S \end{bmatrix}_{N_m\times m} \\ &= \begin{bmatrix} x_{1,1}^s & \cdots & x_{1,d}^s & \cdots & x_{1,m}^s \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1}^s & \cdots & x_{i,d}^s & \cdots & x_{i,m}^s \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N_m,1}^s & \cdots & x_{N_m,d}^s & \cdots & x_{N_m,m}^s \end{bmatrix}_{N_m\times m}, \end{aligned} \tag{5}$$

$$\begin{aligned} C &= \begin{bmatrix} C_1 = X_{N_m+1}^S \\ \vdots \\ C_j = X_{N_m+j}^S \\ \vdots \\ C_{N_c} = X_{N_m+N_c}^S \end{bmatrix}_{N_c\times m} \\ &= \begin{bmatrix} x_{N_m+1,1}^s & \cdots & x_{N_m+1,d}^s & \cdots & x_{N_m+1,m}^s \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{N_m+j,1}^s & \cdots & x_{N_m+j,d}^s & \cdots & x_{N_m+j,m}^s \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N_m+N_c,1}^s & \cdots & x_{N_m+N_c,d}^s & \cdots & x_{N_m+N_c,m}^s \end{bmatrix}_{N_c\times m}, \end{aligned} \tag{6}$$

where M is the mouse population matrix, N_m is the number of mice, C represents the cat population matrix, N_c is the number of cats, and C_j is the fifth cat.

To update the search agents, changing the position of the cats is modeled based on the natural behavior of the cats and moving toward the mice in the first phase. The proposed CMBO mathematical update is modeled using Equations 7 and 8.

$$\begin{aligned} &C_j^{\text{new}} : c_{j,d}^{\text{new}} = c_{j,d} + r \times \left(m_{k,d} - I \times c_{j,d}\right) \text{ and } j = 1 : N_c, \\ &d = 1 : m, k \in 1 : N_m, I = \text{round}\,(1 + \text{round}), \end{aligned} \tag{7}$$

$$C_j = \begin{cases} C_j^{\text{new}} \,|\, F_j^{c,\text{new}} < F_J^c \\ C_j, \,|\, \text{else} \end{cases}, \tag{8}$$

where C_j^{new} is the new position of cat j, $c_{j,d}^{\text{new}}$ is the new value for the variable d obtained by the j^{th} cat, r is a random number in the interval (0, 1), and $m_{k,d}$ represents the dimension d. $F_j^{c,\text{new}}$ is the value of the objective function based on the new position of the j^{th} cat.

In the second step of the proposed CMBO, the escape of mice to shelters is modeled. Our assumption in CMBO is that there is a random shelter for each mouse. The position of the shelters in the search space is randomly determined based on the patterning of the positions of the different individuals in the algorithm. This step of updating the position of mice is mathematically modeled using Equations 9 and 10:

$$\begin{aligned} &M_i^{\text{new}} : m_{i,d}^{\text{new}} = m_{i,d} + r \times \left(h_{i,d} - I \times m_{i,d}\right) \\ &\times \text{sign}(F_i^m - F_i^H) \text{ and } i = 1 : N_m, d = 1 : m, \end{aligned} \tag{9}$$

$$M_i = \begin{cases} M_i^{\text{new}}, \,|\, F_i^{m,\text{new}} < F_i^m \\ M_{i,} \,|\, \text{else} \end{cases}, \tag{10}$$

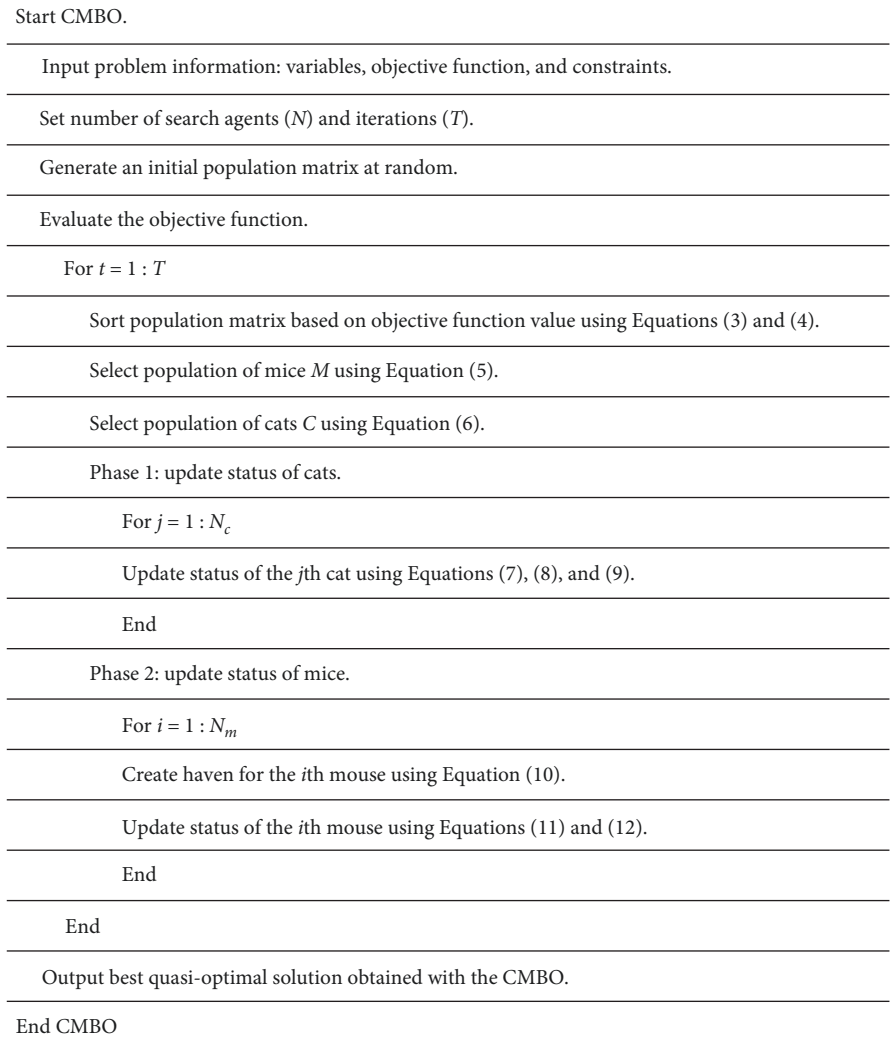

FIGURE 1: Flowchart illustrating the operational steps of the proposed CMBO algorithm.

where h_i is the shelter for the i^{th} mouse and F_i^H is the value of its objective function. M_i^{new} is the new position of the i^{th} mouse and $F_i^{m,new}$ is the value of its objective function.

After updating all individuals of the algorithm population, the algorithm enters the next iteration. According to Equations 5, 6, 7, 8, 9, 10, 11, and 12, the execution of the algorithm is repeated until the stopping condition is reached. The stopping condition of optimization algorithms can be based on a specified number of iterations or a defined acceptable fault between the solutions obtained in successive iterations. In addition, the stopping condition of the algorithm may be a certain period. After completing the iterations and fully executing the algorithm on the optimization problem, CMBO outputs the best quasi-optimal solution obtained. The flowchart and pseudocode of the proposed CMBO are presented in Figures 1 and 2, respectively.

2.3. Proposed Architecture. In Figure 3, our unique approach to implementing the Apriori algorithm is illustrated. In the proposed architecture, two parallel Nvidia graphics cards are used instead of the CPU. Traditionally, the Apriori algorithm is applied on an entire dataset immediately. However, our method proposes applying the Apriori algorithm separately in two halves of the dataset and then concatenating the results at the end. This division and concatenation strategy offers several advantages:

(i) Efficiency in parallel processing: by dividing the dataset into two, we can take advantage of parallel processing capabilities, such as those provided by multicore CPUs or multiple GPUs, to process both halves concurrently, leading to a significant reduction in overall computation time.

(ii) Memory management: large datasets can be memory-intensive when processed in their entirety. By splitting the dataset, we reduce the memory overhead, making the algorithm more manageable and scalable for even larger datasets.

(iii) Enhanced accuracy: by processing the dataset in halves, we may achieve more granular insights and detect patterns that might be overlooked when the dataset is processed as a whole.

(iv) Flexibility in analysis: this approach offers flexibility. For instance, one can compare the results from the two halves to understand any discrepancies or

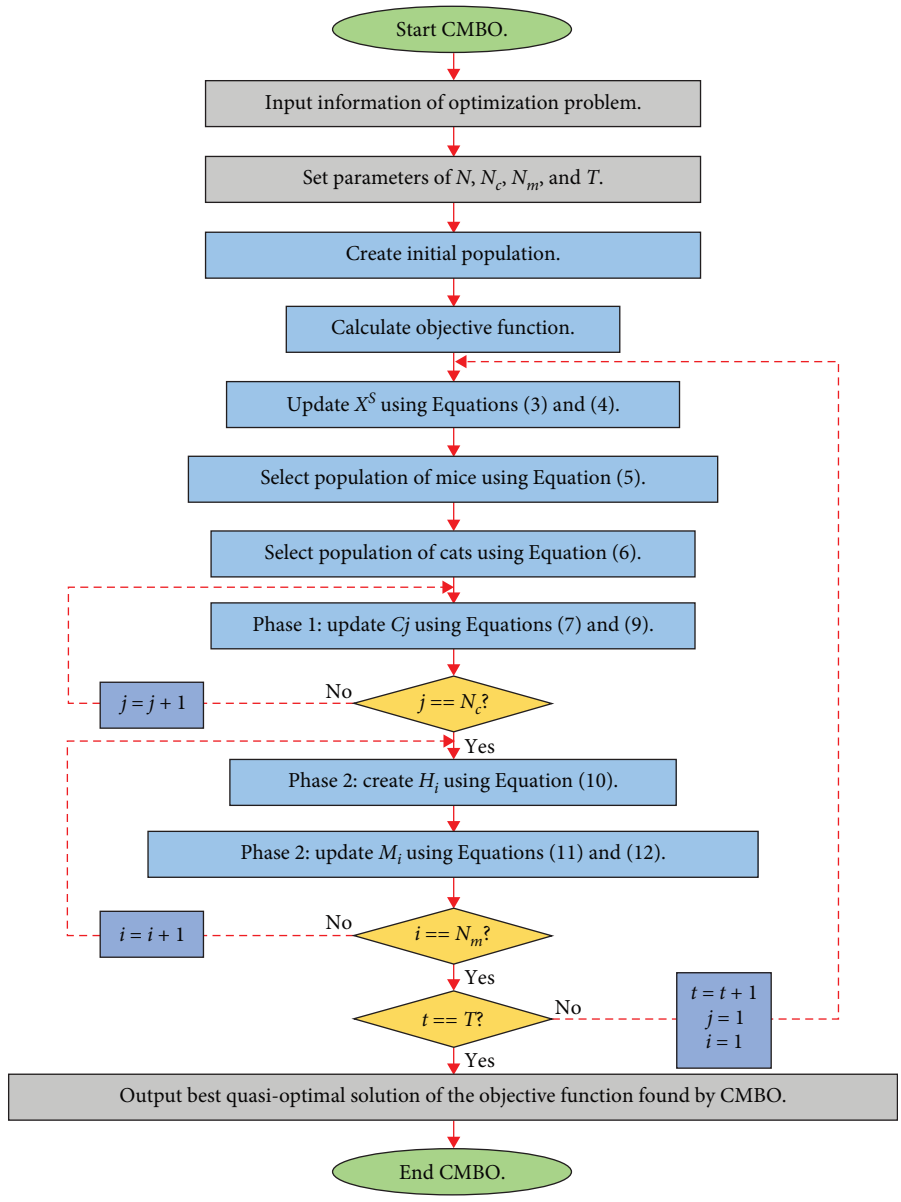

Figure 2: Pseudocode representation of the CMBO algorithm.

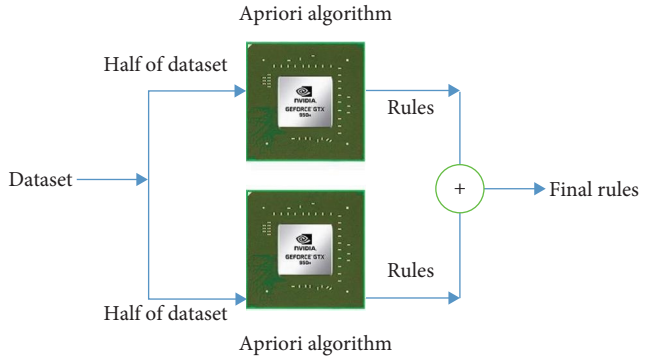

Figure 3: Architectural representation of the parallelized Apriori algorithm using dual Nvidia graphics cards, illustrating data division, concurrent processing, and results concatenation.

anomalies in the data. Moreover, if one half of the dataset has an issue (e.g., corrupted data), it will not hinder the processing of the other half.

(v) Simpler aggregation: once the Apriori algorithm has been applied to both halves, the results can be concatenated. This aggregation step can further be optimized based on the unique patterns derived from each half, ensuring that the final output is comprehensive and represents the entire dataset accurately.

In this research, the similarity evaluation criteria produced in [31] are used. As a result, a consistent set of rules with minimal overlap is created. As mentioned before, support and confidence criteria [31] are defined according to Equations 11 and 12:

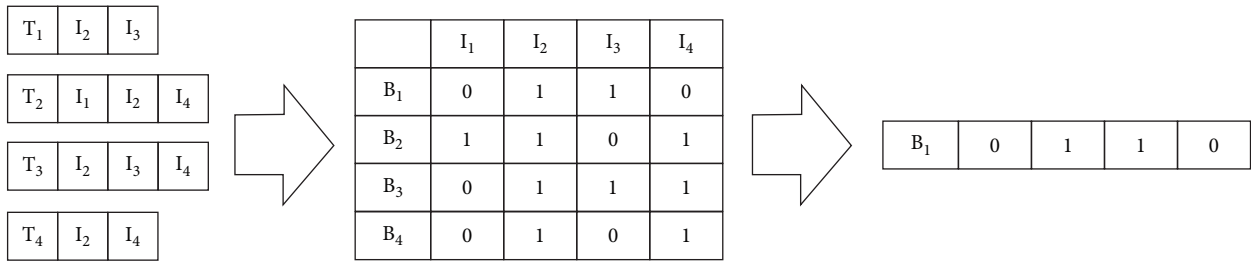

FIGURE 4: An example of moths.

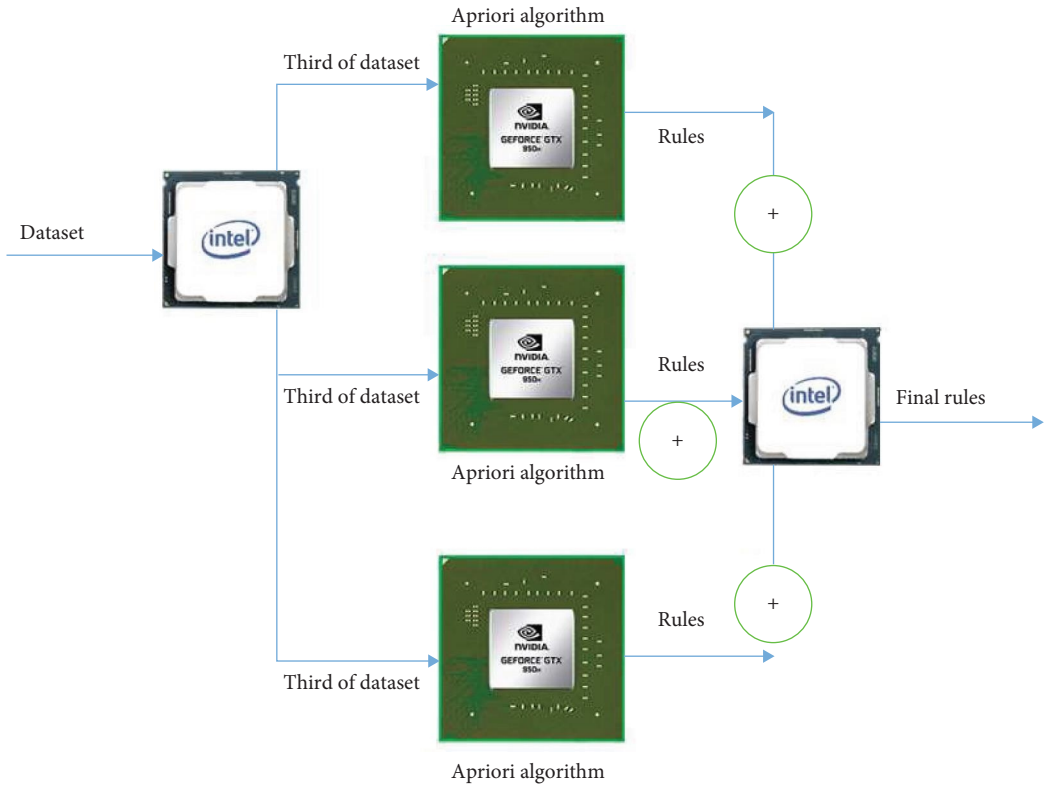

FIGURE 5: Parallel processing scheme with three GPUs.

$$\text{Support}\,(r) = \frac{|X \cup Y|_T}{|T|}, \tag{11}$$

$$\text{Confidence}\,(r) = \frac{\text{Support}\,(X \cup Y)}{\text{Support}\,(X)}. \tag{12}$$

The value of these two defined threshold-based criteria must be greater than the specified threshold value. Also, each particle/moth is defined as follows.

In Figure 4, the first moth B1 has two transactions, I2 and I3, whose values are set to one, and the rest of the entries are defined as zero.

In the algorithm, two first and fourth transactions have been selected for B2, and three transactions (I2, I3, and I4) have been selected for B3. The fourth moth also has only two transactions, I2 and I4. Each moth is a representation of a selection of transactions. The desired system in the proposed architecture is parallelized using three GPUs, as shown in the Figure 3. After the data is fed to the CPU, it is divided into three equal parts, with each part assigned to a GPU. This three-way division was chosen for several reasons. First, dividing data into three parts offers a balance between parallel processing speed and the management overhead associated with data distribution and subsequent integration. Dividing the dataset into more parts, like four or five, might lead to increased management overhead and possible redundancy in the extracted rules, reducing the efficiency of the parallel processing. Second, using three GPUs allows us to maximize resource utilization while ensuring that each GPU has a sufficiently large subset of the data to process, enabling efficient extraction of rules. Last, from a hardware perspective, our system configuration was best suited to handle three GPUs, providing optimal power and thermal performance (Figure 5).

Finally, GPUs extract the rules based on the proposed algorithm and send them to the CPU after the screening with

Table 1: UCI machine learning datasets [32].

Name	Number of transactions	Number of items
Zoo	102	17
Lymphography	148	48
Soybean	683	36
Australian	690	60
Izmir weather	1,461	10
Segment	2,310	19
Splice	3,190	6
Nursery	12,690	9
House 16H	22,784	17
Connect-4	67,557	43

Table 2: UA FIM repository datasets [33].

Name	Number of transactions	Number of items
IBM-Quest-standard	10,000	40
Chess	3,196	75
Mushroom	8,124	119
PumbStar	40,385	7,116
MS-WebView-1	59,602	497
BMS-WebView-2	77,512	3,340
Korasak	80,769	7,116
Retail	88,162	16,469
IBM-Artificial	100,000	999
BMP-POS	515,597	1,657

support and confidence. The CPU applies the union operator to the rules obtained and displays them to the user. The reason for the union operator is the possibility of duplicate offers.

3. Experimental Results

3.1. Setting. To prove the efficiency of running speed, we performed a series of implementations of our proposed scheme under different hardware configurations. First, the scheme was executed solely on "one CPU." Thereafter, we extended the configuration to include "one CPU + two GPUs", harnessing the added computational power of the graphics processing units. Last, the scheme was executed in its full proposed format, optimizing both the hardware and the algorithmic approach.

All these implementations were meticulously coded using MATLAB 2017b, ensuring consistent experimental settings throughout the evaluations. For a comprehensive assessment, we used two distinct datasets to test the efficacy of our method. The first dataset was sourced from the renowned UCI Machine Learning Repository [32], and the second from the UA FIM Repository [33]. Detailed specifications and results pertaining to each dataset's evaluation are presented in Tables 1 and 2, respectively.

In the experiments, we leveraged Nvidia graphics cards and Intel processor. Their technical details are outlined in Tables 3 and 4.

3.2. Model Performance. We conducted a comparative analysis of the CMBO technique against other renowned optimization methodologies, including the genetic algorithm (GA), particle swarm optimization (PSO), artificial bee colony (ABC), and differential evolution (DE). Our findings suggest that CMBO not only outperformed these techniques but also showcased distinct advantages. Specifically, CMBO exhibited faster convergence rates, indicating its efficiency in reaching optimal or near-optimal solutions in a reduced timeframe. Furthermore, its adept handling of local minima ensured that the algorithm did not get trapped in suboptimal solutions, thereby enhancing its robustness and reliability in complex optimization landscapes.

The efficacy of the proposed technique was gauged using three distinct algorithms, namely BSO, GBSO-Miner, and BOA [31]. GBSO-Miner is an innovative approach built upon the BSO framework, wherein every step of BSO, from delineating the search perimeter, conducting a local search, performing evaluations, to the dance sequence, is executed on the GPU. To bridge the gap between each task data input and GPU entities, an intricate mapping technique is employed. An exhaustive array of tests presented in [31] underscores the superior efficiency of the GBSO-Miner platform when juxtaposed with benchmark methodologies from academic literature, such as GPApriroi and MEGPU, particularly in the realm of text and graph database evaluations. Several factors underscore the choice of these algorithms for a comprehensive and fair comparison with the proposed method. These include the swarm-centric essence of both BSO and GBSO-Miner, the GPU-centric parallelism inherent to GBSO-Miner, and its capacity to function seamlessly across a multi-GPU infrastructure spread over a network of nodes. Moreover, past records have illuminated the edge GBSO-Miner holds over other conventional methods documented in academic literature.

Figures 6 and 7 present experimental outcomes derived from the UCI machine learning datasets and the UA FIM repository datasets, in that order. The data in Figure 6 suggests that the method introduced in this study aligns closely with the results yielded by the GBSO-Miner approach in five distinct scenarios. Further analysis of Figure 6 indicates that as transaction volumes escalate, there is a discernible dip in accuracy. This decline can be attributed to the proliferation of local minima, which is a consequence of heightened transaction counts. Additionally, the frequency of iterations and the peak GPU utilization duration emerge as constraints, diminishing the precision of the techniques when handling an extensive volume of transactions. Figure 7 corroborates the insights drawn from Figure 3 but pivots on the UA FIM repository datasets. Given that the transactional volume detailed in Table 2 substantially overshadows that of Table 1, the accuracy of rule extraction dwindles more rapidly. Consequently, when the novel method is deployed, there is an enhancement in the quality, as reflected in the elevated percentage of accurate rule extraction across numerous instances.

In most cases, the proposed algorithm improves the quality regarding the percentage of correct detection of rules and recommendations. Therefore, experiments expose the capabilities and improvements of the algorithm. At the same time, a new architecture was introduced to realize higher processing speed. To demonstrate the quality, the proposed

Table 3: Technical specifications of the Nvidia graphics card used in the experiments.

Property	GeForce GTX 950M	Supports double	1
Compute capability	5.0	Multiprocessor count	5
Available memory	3.5125e + 09	Total memory	4.2950e + 09
Clock rate (KHz)	1,124,000	SIMD width	32
Driver version	8	Tool kit version	8
Max threads per block	1,024	Max shmem per block	49,152
Max thread block size	(64 1024 1024)	Max grid size	(2.1475e + 09 65535 65535)

Table 4: Technical specifications of the Intel processor card used in the experiments.

Name	6th Generation Intel® Core™ i7 processors
Processor number	i7-6700HQ
Launch date	Q3′15
Processor base frequency	GHz 2.60
Max turbo frequency	GHz 3.50
Cache	MB Intel® Smart cache 6
Bus speed	GT/s 8

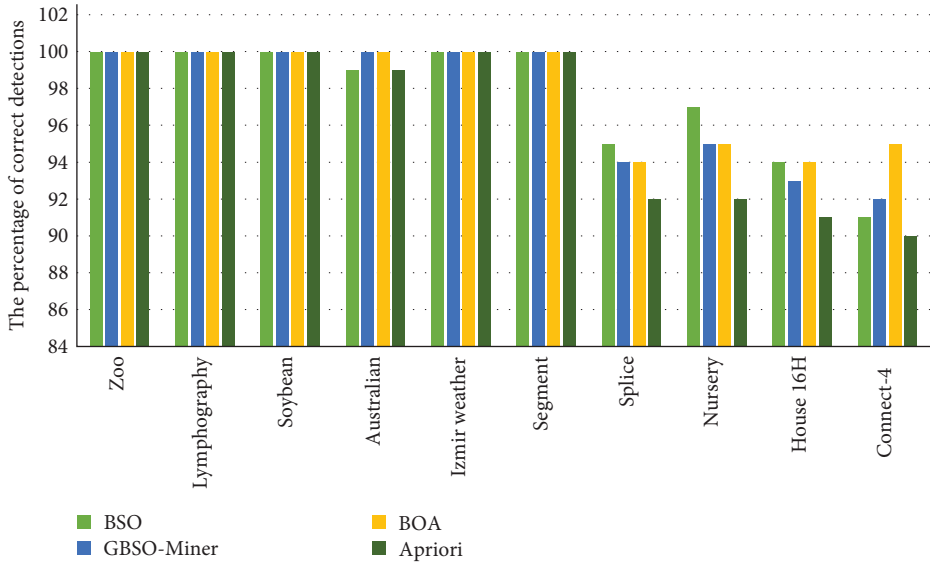

Figure 6: Results on UCI Machine Learning datasets. The results for the methods BSO, GBSO-Miner, and BOA were obtained from [31].

architecture was implemented in "one CPU", "one CPU plus two GPUs", and finally, "one CPU plus three GPUs" on the IBM-Quest-standard dataset, and its speed was evaluated. Different examples of architecture implementation all reached the same quality and 100% accuracy (Table 5). For a better evaluation of the output, the resulting running times are shown in Figure 8.

As shown in Figure 9, the architecture on "one CPU and two GPUs" got a good running time, and the architecture on "one CPU plus three GPUs" got the best running time.

In the above line plot, the difference between different designs is visible. The comparison of the output of the fourth design, i.e. "one CPU plus three GPUs" with "one CPU", is shown in Figure 10. According to the graph, the processing time of the fourth design took about half of the processing time on "one CPU".

4. Discussion

This article proposed a novel approach to address the challenges in extracting frequent patterns from large datasets in the realm of data mining, especially considering temporal and spatial complexities. While recognizing the foundational role of the Apriori algorithm, the study acknowledged its limitations in handling voluminous datasets. To remedy

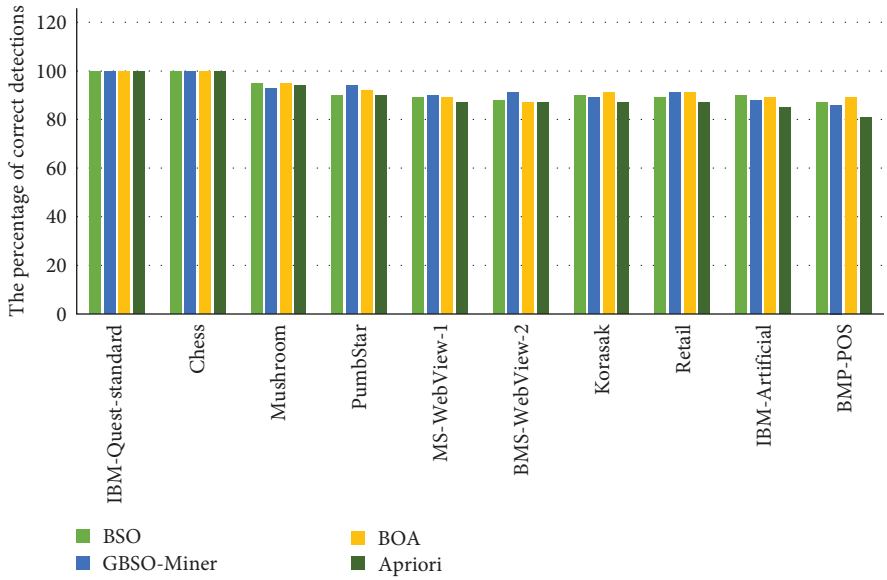

Figure 7: Results on UA FIM repository. The results for the methods BSO, GBSO-Miner, and BOA were obtained from [31].

Table 5: Comparison of the four designs on the dataset.

CPU + 3 GPU (s)	CPU + 2 GPU (s)	CPU + 1 GPU (s)	CPU (s)	Number of transaction
156.1	246.1	354.2	301.1	1,000
436.5	599.7	812.2	786.4	2,000
565.2	772.4	1,379.9	1,371.1	3,000
996.8	1,233.1	1,811.4	1,749.6	4,000
1,809.6	2,317.6	2,805.3	2,707.3	5,000
1,875.1	2,387.3	3,651.9	3,415.4	6,000
2,110.1	3,111.2	4,580	4,429.6	7,000
2,226.1	3,654.4	4,888.7	4,737.1	8,000
2,555.2	3,815.3	5,154.6	5,100.2	9,000
2,789.8	4,190.6	6,017.7	5,841.8	10,000

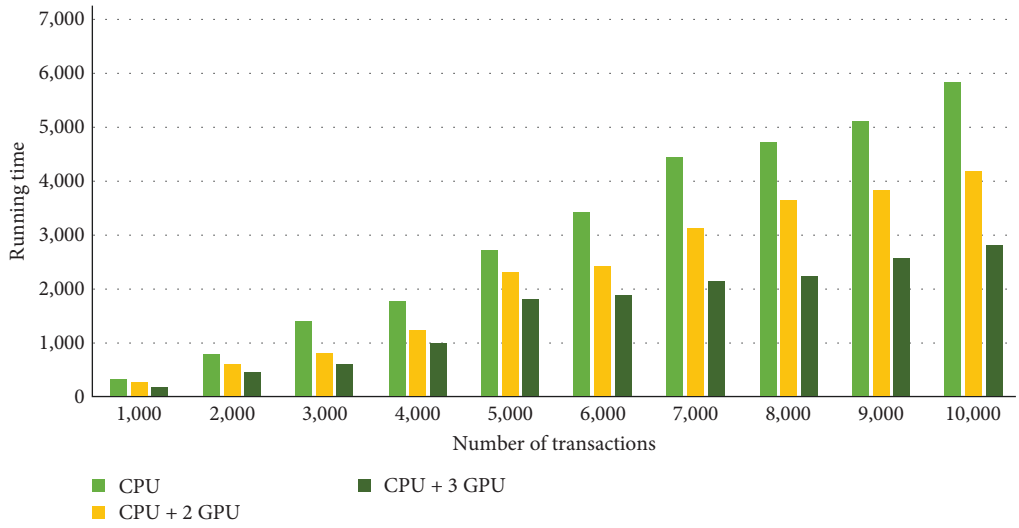

Figure 8: Running time bar plot.

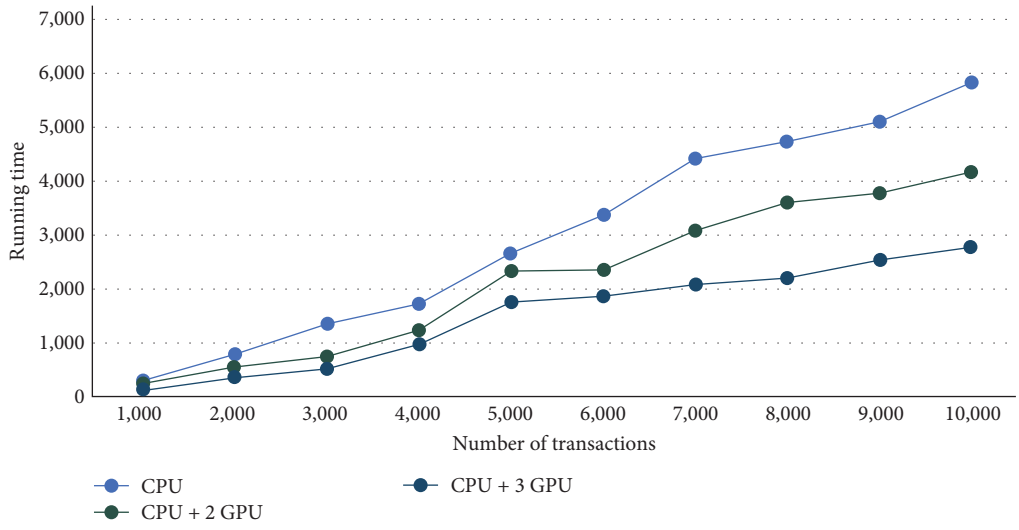

Figure 9: Running timeline plot.

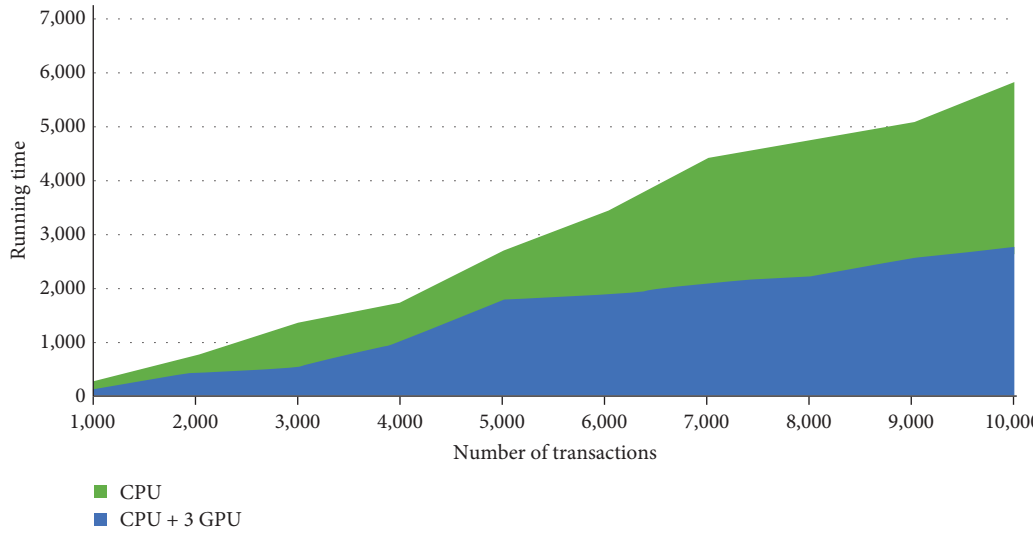

Figure 10: Comparison of the design of "one CPU plus three GPUs" with "one CPU".

this, the research unveiled an innovative method harnessing the power of parallel network topologies and GPU architectures. Central to this proposal were two main elements: first, the employment of parallel processing to facilitate faster attainment of ideal outcomes, and second, integrating the CMBO algorithm. The latter, inspired by the natural chase dynamics between feline predators and their rodent prey, operates on a two-stage paradigm: an initial chase phase led by the cats followed by a strategic evasion stage orchestrated by the mice. Agents in this model were further distinguished based on their objective function evaluations. Additionally, the proposed system design effectively incorporated dual Nvidia graphics cards in tandem, offering a performance edge over traditional CPU setups. Collectively, the advanced method not only addressed the shortcomings of the Apriori algorithm but also augmented the efficiency in deducing association rules and discerning recurrent patterns with heightened accuracy. An exhaustive assessment across diverse network topologies was undertaken, explaining the strengths and weaknesses of each. When juxtaposed with the Apriori benchmark, the proposed technique exhibited superior performance both in rapidity and proficiency, marking a pivotal advancement in data mining.

Given the complexities and requirements of our problem, we sought an optimization technique that could aptly model intricate behaviors. The CMBO, inspired by the natural behaviors of cats and mice, offered a novel approach to this. Its two-phase movement modeling, cats moving towards mice and mice evading to safety, aligned well with our problem dynamics. Additionally, the adaptability of CMBO to diverse environments and its balance between exploration and exploitation provided an edge over traditional optimization methods. As cats naturally possess the ability to adjust their hunting strategies based on the behaviors of mice, this flexibility ensured that the model could dynamically adjust its search pattern in the solution space. Moreover, the interaction between cats and mice inherently leads to a continuous refinement of strategies

on both ends, making it a robust mechanism to avoid local optima. By leveraging this dual-aspect movement of attraction and evasion, the CMBO ensures that potential solutions are not just local but are examined in a global context. Additionally, the diversity exhibited by different cats and mice in their hunting and evasive strategies adds a unique element of parallel search pathways. This diversity further ensures that the optimization does not stagnate, and there is always a fresh perspective to approach the problem. Furthermore, the natural competitiveness among the cats to catch the mice and the urgency of the mice to evade capture introduces a time-sensitive dynamic, making the CMBO more efficient in finding an optimal or near-optimal solution in shorter time frames.

As association rule mining on GPUs gains prominence, it becomes crucial to address the conspicuous absence of the Apriori algorithm's implementation on GPU with CUDA and the GBSO-RSS algorithm within this research. Each proposed methodology carries inherent merits, suggesting potential advances in computational efficiency and accuracy. The transformative nature of GPU technology has undeniably reshaped the data processing paradigm, propelling GPUs to the forefront of high-performance computing. However, this investigation was deliberately tailored to delve deep into the confluence of cat-and-mouse optimization and the Apriori algorithm. Our objective was to navigate the myriad benefits this hybrid approach can offer, thus contributing a nuanced perspective to association rule mining. While the benefits of GPU-centric computations are substantial, one must be wary of the complexities they introduce. Potential hurdles, such as managing memory limitations and navigating specialized programing frameworks, can often temper the enthusiasm of direct adoption. While the GBSO-RSS's robust capabilities are laudable, its integration did not align seamlessly with the research's foundational goals. It is pertinent to recognize that in the dynamic realm of association rule mining, no singular technique can assert absolute dominance. This opens avenues for future research to weave together the capabilities of various methodologies, potentially ushering in a new epoch marked by unmatched computational efficacy and precision in association rule mining pursuits.

The preference for the GTX 950M over more recent GPUs, such as Nvidia's RTX series which boasts notably enhanced performance metrics over its predecessor, is rooted in several considerations. Foremost, the GTX 950M offers an ideal balance between performance and cost-efficiency, establishing it as a favored option for a broad range of researchers and professionals. Although RTX GPUs offer undeniable performance superiority, they come with a steeper learning curve due to their avant-garde technologies and features—some of which may exceed the immediate needs of our study. It was also essential to align with previous studies, many of which potentially employed the more widespread GTX 950M GPU. That said, the cutting-edge functionalities of RTX GPUs underscore a valuable direction in data mining, meriting thorough investigation in future research pursuits.

The decision to employ the 6th Generation Intel Core i7 Processors, instead of more recent options like the 13th Generation Intel Core i7, was influenced by several factors. First and foremost, the reliability and performance of the 6th Generation Intel Core i7 have been well-established, and it effectively serves the computational requirements of our study without incurring undue expenses. In addition, by selecting this processor, we strived to ensure our research remains congruent with existing benchmarks, fostering both compatibility and the ability to be juxtaposed with a wide range of prior research that predominantly relies on this generation. It's also worth noting that transitioning to cutting-edge hardware, such as the 13th Generation Intel Core i7, might bring about unexpected challenges, from navigating new configurations to potential compatibility concerns, which could detract from our primary research objectives.

The model we proposed emerges as a multifaceted instrument, holding promise for a wide array of image processing undertakings. Its efficacy becomes especially salient in the realm of image retrieval systems, where it acts as a potent preprocessing conduit. Existing literature encompasses myriad pivotal studies that align seamlessly with our approach. Notably, the work presented in [34] ventures into the domain of intricate image reconstruction methodologies. Such methodologies could derive substantial advantages from sophisticated preprocessing paradigms, akin to the one we introduce. Similarly, the study highlighted in [35] provides a comprehensive examination of the intricate relationship between texture and color features, pivotal in image retrieval processes. The distinct attributes of our proposed model stand in a position to synergize with and augment the insights from these studies. In concert, these investigations serve not only as a testament to the significance and potential of our model but also chart a promising course for ensuing research endeavors, underscoring the model's adaptability to a diverse range of application spheres.

5. Conclusion

The expansion of the web and the proliferation of available information have made it necessary to have appropriate tools for data classification, providing users with desired data, and changing the type of information provided based on the needs of users. The purpose of extracting information from web data is to facilitate the use of the web by users, quick and easy access to information, and help designers and information providers to supply the best services to users with the least cost and get the most benefits. None of these objectives can be achieved except through an automatic information extraction system or a recommender system. In this study, a new algorithm based on CMBO was presented to detect suggested items with better quality than other samples. The results related to the detection percentage of the correct rules by the CMBO method showed that the quality of the algorithm is better or equal to the classic methods including Apriori, BSO, and GBSO-Miner.

In addition, the test results proved that processing with "three parallel GPUs under one CPU" is much faster than processing with "one CPU". According to the displayed graphs, the processing time of the proposed architecture is about half of the processing time on "one CPU". Therefore,

in the first stage, it is quite evident that the processing speed of the proposed architecture is higher. The time complexity is divided by the number of GPUs. Also, the time complexity of the cat-and-mouse optimization algorithm is n^2. The memory required for the cat-and-mouse algorithm is negligible. The memory required for the parallelization part is almost the same as the memory required in "one CPU". One of the limitations of this study is the difficulty of networking GPUs and the need for a special model for this purpose. The use of more GPUs in the cloud computing platform is suggested for future work.

In future work, we envision delving deeper into the integration of deep learning techniques to further revolutionize the extraction of frequent patterns from extensive datasets. Deep learning, with its unparalleled capacity to handle vast amounts of unstructured data and unearth intricate patterns, offers promising avenues to amplify the capabilities of our current model. Neural networks, particularly convolutional and recurrent architectures, could be adeptly tailored to handle the temporal and spatial dimensions of data, offering more nuanced insights into pattern recognition. Additionally, the fusion of CMBO with deep learning models might enable the algorithms to not just mimic instinctual dynamics, but also learn and adapt from the data itself, ensuring more robust and adaptable solutions. Exploring the synergy between GPU-accelerated parallel processing and deep learning architectures will be central to this endeavor, aiming to achieve unprecedented speeds and accuracies in frequent pattern extraction. Ultimately, by merging the power of CMBO and deep learning, we aspire to establish a novel paradigm in data mining, one that seamlessly marries intuition with computational prowess.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

There are no conflicts of interest to report.

Authors' Contributions

All co-authors have seen and agree with the contents of the manuscript.

References

[1] S. G. Alonso, I. De La Torre-Díez, S. Hamrioui et al., "Data mining algorithms and techniques in mental health: a systematic review," *Journal of medical systems*, vol. 42, no. 9, Article ID 161, 2018.

[2] S. Zhu, "Research on data mining of education technical ability training for physical education students based on Apriori algorithm," *Cluster Computing*, vol. 22, no. Suppl 6, pp. 14811–14818, 2018.

[3] E. Baralis, T. Cerquitelli, S. Chiusano, and R. Meo, "Data mining in databases: languages and indices," in *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, pp. 341–351, Springer, Cham, 2018.

[4] B. Bakhshinategh, O. R. Zaiane, S. ElAtia, and D. Ipperciel, "Educational data mining applications and tasks: a survey of the last 10 years," *Education and Information*, vol. 23, no. 1, pp. 537–553, 2018.

[5] A. Pavithra and S. Dhanaraj, "Comparative study of effective performance of association rule mining in different databases," *Data Mining and Knowledge Engineering*, vol. 10, no. 4, pp. 74–77, 2018.

[6] F. Zhang, Y. Zhang, and J. Bakos, "GPApriori: GPU-accelerated frequent itemset mining," in *2011 IEEE International Conference on Cluster Computing*, pp. 590–594, IEEE, Austin, TX, USA, 2011.

[7] G. D'Angelo, S. Rampone, and F. Palmieri, "Developing a trust model for pervasive computing based on Apriori association rules learning and Bayesian classification," *Soft Computing*, vol. 21, no. 21, pp. 6297–6315, 2017.

[8] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[9] B. Wang, K. E. Merrick, and H. A. Abbass, "Co-operative coevolutionary neural networks for mining functional association rules," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1331–1344, 2017.

[10] S. Vojíř, V. Zeman, J. Kuchař, and T. Kliegr, "EasyMiner. eu: web framework for interpretable machine learning based on rules and frequent itemsets," *Knowledge-Based Systems*, vol. 150, pp. 111–115, 2018.

[11] Z. Li, L. Li, K. Yan, and C. Zhang, "Automatic image annotation using fuzzy association rules and decision," *Multimedia Systems*, vol. 23, no. 6, pp. 679–690, 2017.

[12] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *Mining Association Rules Between Sets of Items in Large Databases*, vol. 22, no. 2, pp. 207–216, 1993.

[13] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," *Advances in Knowledge Discovery and Data Mining*, vol. 12, no. 1, pp. 307–328, 1996.

[14] J. S. Park, C. Ming-Syan, and S. Y. Philip, "An effective hash-based algorithm for mining association rules," *Acm Sigmod Record*, vol. 24, no. 2, pp. 175–186, 1995.

[15] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data engineering*, vol. 12, no. 3, pp. 372–390, 2000.

[16] A. Savasere, E. R. Omiecinski, and S. B. Navathe, "An efficient algorithm for mining association rules in large databases," Georgia Institute of Technology, 1995.

[17] Z. H. Deng and S.-L. L.v, "Fast mining frequent itemsets using nodesets," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4505–4512, 2014.

[18] F. Fumarola, P. F. Lanotte, M. Ceci, and D. Malerba, "CloFAST: closed sequential pattern mining using sparse and vertical id-lists," *Knowledge and Information Systems*, vol. 48, no. 2, pp. 429–463, 2016.

[19] J. Dongre, G. L. Prajapati, and S. V. Tokekar, "The role of Apriori algorithm for finding the association rules in Data mining," in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 657–660, IEEE, Ghaziabad, India, 2014.

[20] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, 2011.
[21] W. Fang, M. Lu, X. Xiao, B. He, and Q. Luo, "Frequent itemset mining on graphics processors," in *Proceedings of DaMon*, pp. 34–42, Association for Computing Machinery, 2009.
[22] C. Silvestri and S. Orlando, "GPUDCI: exploiting gpus in frequent itemset mining," in *Proceedings of PDP*, pp. 416–425, IEEE, 2012.
[23] P. Fournier-Viger, J. C.-W. Lin, B. Vo, T. T. Chi, J. Zhang, and H. B. Le, "A survey of itemset mining," *WIREs Data Mining and Knowledge Discovery*, vol. 7, 2017.
[24] N. Holden and A. A. Freitas, "A hybrid PSO/ACO algorithm for discovering classification rules in data mining," *Journal of Artificial Evolution and Applications*, vol. 2008, pp. 1–11, 2008.
[25] F. E. B. Otero, A. A. Freitas, and C. G. Johnson, "A new sequential covering strategy for inducing classification rules with ant colony algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, pp. 64–76, 2013.
[26] L. Yang, K. Li, W. Zhang, and Z. Ke, "Ant colony classification mining algorithm based on pheromone attraction and exclusion," *Soft Computing*, vol. 21, pp. 5741–5753, 2017.
[27] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "Hybrid ant colony optimization and iterated local search for rules-based classification," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 4, pp. 657–671, 2020.
[28] Y. Djenouri, D. Djenouri, A. Belhadi, P. Fournier-Viger, J. C.-W. Lin, and A. Bendjoudi, "Exploiting GPU parallelism in improving bees swarm optimization for mining big transactional databases," *Information Sciences*, vol. 496, pp. 326–342, 2019.
[29] X. Zhao, Y. Fang, S. Ma, and Z. Liu, "Multi-swarm improved moth-flame optimization algorithm with chaotic grouping and Gaussian mutation for solving engineering optimization problems," *Expert Systems with Applications*, vol. 204, Article ID 117562, 2022.
[30] Á. M. García-Vico, C. J. Carmona, P. González, and M. J. del Jesus, "A distributed evolutionary fuzzy system-based method for the fusion of descriptive emerging patterns in data streams," *Information Fusion*, vol. 91, pp. 412–423, 2023.
[31] A. A. Zoraghchian, M. K. Sohrabi, and F. Yaghmaee, "Exploiting parallel graphics processing units to improve association rule mining in transactional databases using butterfly optimization," *Cluster Computing*, vol. 24, no. 4, pp. 3767–3778, 2021.
[32] D. Dua and C. Graff, *UCI Machine Learning Repository Irvine*, University of California, School of Information and Computer Science, CA, 2019.
[33] "Frequent itemset mining dataset repository," 2019.
[34] X. Yu, T. Bicer, R. Kettimuthu, and I. Foster, "Topology-aware optimizations for multi-gpu ptychographic image reconstruction," in *Proceedings of the ACM International Conference on: Supercomputing*, pp. 354–366, Association of Computing Machinery, 2021.
[35] M. K. Kelishadrokhi, M. Ghattaei, and S. Fekri-Ershad, "Innovative local texture descriptor in joint of human-based color features for content-based image retrieval," *Signal, Image and Video Processing*, vol. 17, no. 8, pp. 4009–4017, 2023.