International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 OPEN ACCESS
Available Online at : www.ijsrcseit.com
doi : https://doi.org/10.32628/CSEIT2390559

A Secure and Optimization Based Clustering for Vertical and Horizontal Fragmentation in Distributed Database Management System

D. Sahithi[1], **Dr J. Keziya Rani**[2]
[1]Research Scholar, Department of Computer Science & Engineering, S.K University, Anantapuramu, Andhra Pradesh, India.
[2]Assistant Professor, Department of Computer Science & Engineering, S.K University, Anantapuramu, Andhra Pradesh, India

ARTICLEINFO

Article History:

Accepted: 15 Dec 2023
Published: 30 Dec 2023

Publication Issue
Volume 9, Issue 6
November-December-2023
Page Number
322-332

ABSTRACT

Distributed database systems are increasingly important due to the massive data output, and their effectiveness is largely based on their design. Two key processes, fragmentation and allocation, are used to improve the efficiency and efficacy of these systems. Effective data fragmentation requires both horizontal and vertical categorization of tuples. Advanced optimization techniques are used for both fragmentations, such as the Enhanced Arithmetic Optimization (EAO) algorithm with Opposition-based Learning (OBL) and Levy Flight Distributer (LFD) for vertical fragmentation and the hybrid Aquila Optimizer (AO) with Artificial Rabbit Optimization (ARO) algorithm for horizontal fragmentation. The fragmented data is securely transmitted using the Fully Homomorphic Encryption (FHE) algorithm. The implementation is executed using the Python language, and the performance of the proposed algorithms is evaluated using different performance parameters. The execution time analysis shows that the proposed EAOA algorithm consumes 5.5 seconds for vertical fragmentation, while the hybrid AARO algorithm takes 5.9 seconds for horizontal fragmentation. The vertical fragmentation is found to be better than the horizontal one in DDBMS.

Index terms: Vertical Fragmentation, Horizontal Fragmentation, Distributed Database Processing, Attributes, Optimization-Based Clustering, Security, And Execution Time Analysis

I. INTRODUCTION

Database Management System (DBMS) is a computer program that is designed to manage data stored in a relational database [1]. It is a critical component for any organization, as it allows the user to organize, store, edit, and retrieve data with relative ease. DBMSs provide several features such as security, data integrity, scalability, data backup, data recovery, and database maintenance [2, 3]. DBMSs can also help organizations in incorporating relevant data into their business intelligence or analytics platforms. The basic components of DBMSs include transaction management, structural access control, data manipulation, query processing, data definition, and relational algebra [4-7]. It is essential to work on the data stored in to DBMSs; hence DBMSs use the concepts of ACID compliance to ensure the integrity of data in the database. ACID stands for atomicity, consistency, isolation, and durability [8, 9].

The fragmentation model partitions databases into smaller subsets for distributed across multiple sites or machines[10-13]. It allows horizontal and vertical partitioning, reducing response time and providing fault tolerance by eliminating single points of failure. This model also ensures system reliability and prevents unexpected changes from crashing[14-18].

A novel optimization-based fragmentation approach has been proposed to address vertical data fragmentation in databases. This approach focuses on optimal physical layout and query evaluation performance, involving column fragmentation strategy and sharding policies. A cost-based analysis system balances computational resources, memory usage, write and read performance, identifying the best fragmentation strategy and sharding policy for the application[19-22]. The major contributions of this research are given below:

- To perform an enhanced optimization-based clustering for vertical fragmentation
- To develop a hybrid optimization-based clustering algorithm for horizontal fragmentation.
- To develop a lightweight security algorithm for maintaining security while the distribution of fragmented data
- To validate and compare performances by means of execution time, communication cost and complexity analysis for showing which fragmented method provides better results.

The remaining sections of the research paper are organized as follows: the existing literatures related to the proposed methodology is reviewed in section 2. The detailed explanation about the proposed methodologies for both the fragmentation process is described in section 3. The simulation result analysis and its outcomes are depicted in section 4. Finally, the overall conclusion of the research paper is detailed in section 5.

II. RELATED WORKS

Some of the existing methods related to this research are reviewed here.

Mehta et.al. [23] proposed the optimal vertical fragmentation of distributed databases using the differential bond energy algorithm, which determines the optimal partition point with less delay, but is better for high-dimensional data. Hanan et.al. [24] utilized block chain technology for big data security, employing techniques like fragmentation, encryption, and access control, considering data owner requests and minimizing encryption overhead for sensitive data sections. Basim et.al. [25] introduced the BVAGQ-AR algorithm for fragmented database replication management, focusing on synchronous replication and splitting databases into distinct chunks. However, it faces limitations in handling statement failure and application software errors.

Yong et.al. [26] used the distributed memetic algorithm to improve database fragmentation, enhancing privacy and utility. The method uses a random distributed framework, dynamic grouping recombination operator, mutation operators, and two-dimension selection approach. An efficient heuristic horizontal fragmentation is presented and allocation technique developed by Ali A.Amer et al. [27]. It integrates these tactics into a unified strategy, resulting in a significantly enhanced solution for DDBS productivity promotion. The detailed illustrations of both internal and external evaluations have been documented to improve DDBS performance. Mehdi Goli et al. [28] proposed a distributed swarm intelligence algorithm for vertical fragmentation, minimizing operating costs and maximizing localization of transaction processing. Adel A. Sewisy et al. [29] designed a heuristic query-driven clustering-based vertical fragmentation approach, aiming to cultivate query clusters and avoid discontinuous pieces in dynamic and static DDBMS environments. Both methods outperform others in terms of computational complexity and result optimality.

III. Proposed Methodology

The main objective of this study aims to create a database fragmentation model that supports horizontal and vertical partitioning in distributed environments. The model ensures safe, coherent tuple-based sub-databases, enhancing data retrieval efficiency and ensuring efficient dispersion over multiple servers.

The research introduces an improved version of the ***Enhanced Arithmetic Optimization Algorithm (EAOA)*** to improve the vertical fragmentation strategy. The AOA, which has been successful in global issues, is combined with Opposition-based learning (OBL) and Levy flight distribution (LFD) to address its shortcomings in challenging and high-dimensional situations..

The hybrid Aquila Optimizer (AO) with ***Artificial Rabbit optimization (ARO)***-based clustering technique is used for horizontal fragmentation of input data. The fragmented data is securely transmitted using Fully Homomorphic Encryption (FHE), obtained separately. The decryption procedure evaluates the encryption's effectiveness for both horizontal and vertical fragmented data. Data from a dispersed environment is processed, and performance comparisons are conducted to determine the best fragmentation model for distributed environments..

3.1 Vertical fragmentation

In a distributed database, vertical fragmentation is the act of breaking out a table vertically based on columns or properties. A system stores some of its properties in vertical fragmentation, while other systems store the remaining attributes. This is because not every website will use all of a table's columns. Data privacy may be protected by using vertical fragmentation. In order to allow most applications to operate in a single fragment, vertical partitioning aims to divide a connection into a collection of smaller relations. The next subsections explain the vertical fragmentation using EAOA with proper explanation.

3.1.2 Enhanced AOA

In contrast to just executing S or A, OBL is dedicated to considering both candidate solutions and their opposing solutions, which indicates a better possibility to attain the global optimal and quicker acceleration of convergence. It is used to locate a solution that is the opposite of the current solution, and then it compares the values of their fitness functions to see if the opposite solution is being employed. For example, $x*(C_{iter})$ is saved if $f(x*(C_{iter})) \leq f(\bar{x}*C_{iter})$),; if not, $\bar{x}*(C_{iter})$ is stored. To obtain the opposite answer in OBL, the equation is given below.

$$\bar{x}*(C_{iter}) = ub + lb - x*(C_{iter}) \tag{1}$$

where the best solution's location in the current iteration is indicated by the symbol $x*(C_{iter})$. The opposite location of the best solution in the current iteration is indicated by the symbol $\bar{x}*(C_{iter})$. The LRS, which creates a random jump using the Lévy distribution, is incorporated into the updated candidate solutions, which are also generated based on the arithmetic operators of the subset of solutions as in AOA. This allows the candidate solution to randomly move to a new location in order to search for better candidate solutions in each iteration. The parameters of the population size (N), the decision variables (Dim) depending on the problem's dimension, and the LRS define the jump's magnitude and direction. Finding the best answer requires a bit more time with LRS's greater search space capability than with the original AOA. With the help of the LRS (S), the search area of exploration in (2) and the exploitation in (3) in the projected Levy function are altered.

$$a_{ij}(C_{iter}+1) = \begin{cases} best(a_j) \div S(MoP_{new}+\varepsilon)\left((ub_j - lb_j) \times \mu + lb_j\right), r_2 < 0.5 \\ best(a_j) \times S(MoP_{new}) \times \left((ub_j - lb_j) \times \mu + lb_j\right), \ \ otherwise \end{cases} \tag{2}$$

$$a_{ij}(C_{iter}+1) = \begin{cases} best(a_j) - S(MoP_{new})\left((ub_j - lb_j) \times \mu + lb_j\right), r_3 < 0.5 \\ best(a_j) + S(MoP_{new}) \times \left((ub_j - lb_j) \times \mu + lb_j\right), \ \ otherwise \end{cases} \tag{3}$$

To increase the diversity of the potential solutions and prevent becoming trapped in local optima, equations (2) and (3) are utilized. The suggested technique can possibly discover a new candidate at each iteration that is not identified by the conventional AOA by enabling Lévy Random Steps (LRS) character to explore new sections of the search space. First, the decision variable, restrictions, and objective function are used to determine the fitness function and the optimal solution. Because Lévy Random Steps can enhance the algorithm's capacity for global search and raise the probability of discovering the global optimum, they can result in a more reliable and effective optimization process. In complicated optimization issues, where it may be necessary to investigate several local optima in order to identify the global optimum in which this method can also be helpful. Algorithm 1 displays the pseudo-code of EAOA for vertical fragmentation.

Algorithm 1: Pseudo-code of EAOA for vertical fragmentation

```
Initialize the parameters like $\alpha, \mu$ and initialize the solution's positions randomly.
Initialize fitness function and best solution for the given solutions.
While $(iteration\ Count < Max\ iteration)$ do:
```

```
Update MoA and MoP using equations
for $i = 1$ to $N$ solutions do:
  for $j = 1$ to positions do:
    Generate random variables $r_1, r_2, r_3$ in [0,1]
    if $r_1 > MoA$ and $r_2 < 0.5$
      Apply division using the first logic of equation using (2)
    else if $r_1 > MoA$ and $r_2 < 0.5$
      Apply multiplication using the second logic of equation using (2)
    else if $r_1 \leq MoA$ and $r_3 < 0.5$
      Apply Subtraction using the first logic of equation using (3)
    else if $r_1 \leq MoA$ and $r_3 \geq 0.5$
      Apply Addition using the first logic of equation using (3)
    end if
  end for
  Calculate the fitness function for updated solutions.
  Find the opposite solution of the best solution using equation (1).
  Calculate the fitness function of both current and its opposite solutions using.
  Set the better one as $x * \ (C_{iter} + 1)$
end for
Increase iteration count
end while
Return the best solution.
```

3.2 Horizontal fragmentation

The technique of separating a table horizontally by assigning a relation to each row (or set of rows) is known as horizontal fragmentation. Afterwards, distinct locations within the distributed system can be allocated to these pieces. So, in this research, hybrid AARO algorithm is developed to perform the horizontal fragmentation for the utilized datasets.

3.2.1 Hybrid AARO algorithm

The program mimics aquila predatory behavior during AO exploration, updating search agent locations. However, local optima and poor escape mechanisms hinder its effectiveness. The ARO algorithm has strong local utilization potential, but early defects hinder its exploitation. Algorithm 2 displays the pseudo-code of the suggested hybrid AARO algorithm.

Algorithm 2: Pseudo-code of the proposed Hybrid AARO algorithm

```
Initialize the population size N and the maximum iterations T.
Initialize the location of each search agent $X_i (i = 1, 2, \ldots, N)$.
While $t \leq T$
  Check if the position goes beyond the search limits and adjust it.
  Compute the fitness value of all search agents.
  Set $X_{best}$ as the best solution obtained so far
  for each $X_i$
    Calculate the fitness function.
    if $F(x) \geq 1$ then //exploration of AO
      if $rand \leq 0.5$ then
        Update the search agent's position
      else
        Update the search agent's position
      end if
    else
      Estimate the energy factor E
      if $E > 1$
        Update search agents position // Detour foraging of ARO
      else
        Update search agents position // Random hiding of ARO
      end if
    end if
  end for
  Perform $t = t + 1$
end while
Return $X_{best}$
```

3.3 Security in fragmentation

Data and information systems must be protected from abuse, illegal access, and alteration by means of security. Distributed database security is concerned with safeguarding data from individuals or harmful software. The four primary security components of a distributed system are multi-level access control, authorization, encryption, and security authentication. Encryption is used in this study to safeguard the two fragmented data sets for processing later on. It is a method of encoding data such that only individuals with the proper authorization may decode it. The FHE method is suggested in this study to carry out encryption in both fragmented data; a thorough explanation of the algorithm is provided in section 3.3.1.

3.3.1 FHE algorithm

Let HE be a homomorphic encryption scheme. If HE is an FHE scheme, then it is the set of the augmented decryption circuit (DH E) and the decryption circuit (H E) are in the allowed circuit set CH E. The six phases of an effective FHE technique are Setup, KeyGen, Encrypt, Decrypt, Add, and Multi. Dowlin [30] devised this approach, which is explained as follows:

- $FHE.Setup$ (1^n): Input the security parameter $n = 2^k$, where $k \in Z^+$. Then choose a sufficiently large prime modulus q , where $q \, mod \, n \equiv 1$. The modulus t satisfies the condition that $1 < t < q$, where t is a small plaintext modulus. Coefficients of distributions χ_{key} and χ_{err} are randomly chosen from $\bar{\psi}_\beta$. Let $params(n, q, t, \chi_{key}, \chi_{err})$.
- $FHE.KeyGen$ $(params)$: Input the parameter $params$, choose f' and g from the key distribution χ_{key} randomly, namely $f', g \leftarrow \chi_{key}$. Set $f = [1 + tf']_q$, generate a new f if it has not the inverse element f^{-1}. Generate vectors $\vec{e}, \vec{s} \in R^l$ randomly, and each component is chosen from χ_{err}. Set, $\gamma = [powers \ of \ 2(f) + \vec{e} + \overrightarrow{hs}]_q$, where $h = [tgf']_q$. Output the public key $pk = h$, the secret key $sk = f$ and the evaluation key $k = \gamma$.
- $FHE.Encrypt$ $(pk, m\epsilon R_t)$: To encrypt a plaintext m, generate the ciphertext as follows:

$$c = [\left\lfloor \frac{q}{t} \right\rfloor m + e_2 + hs_2]_q$$

Where, error terms e_2 and s_2 are randomly chosen from χ_{err}.

- $FHE.Decrypt$ (sk, c) : Given the ciphertext c and the secret key sk, the plaintext m can be recovered as follows:

$$m = [\lfloor t/q . [fc]_q \rceil]_t$$

- $FHE.Add$ $(c1, c2)$: Given two ciphertexts $c1$ and $c2$, the fresh ciphertext $cadd = [c1 + c2]q$.
- $FHE.Multi$ $(c1, c2)$: Given two ciphertexts $c1$ and $c2$, firstly compute $c_{temp} = [\lfloor t/q . (c1 \cdot c2) \rceil]q$, then switch c_{temp}'s secret key as follows

$$c_{mul} = \left[< Bit \ Decomp\left(c_{temp}\right), evk > \right]_q$$

Also, Gentry's idea of raising the modulus is used to switch the secret key.

4. Simulation results and analysis

The proposed method, tested using Python 3.7 on an 8GB RAM-equipped PC, demonstrated its efficacy in terms of execution time, query processing time, error function, and security algorithm encryption and decryption times. The fragmentation procedure was carried out using three different datasets, and the results were analyzed and discussed in detail..

4.1 Dataset description

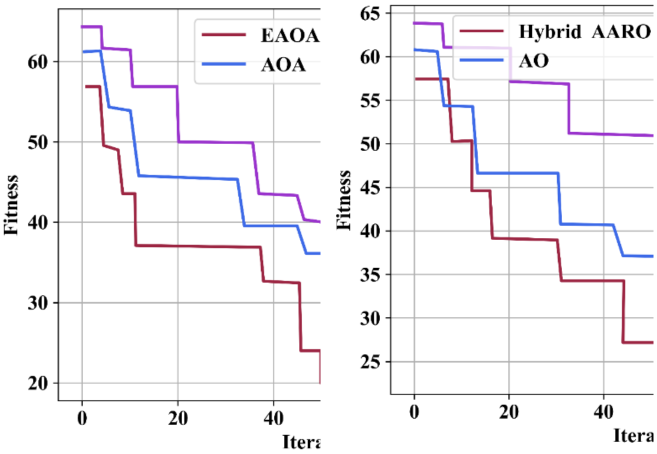
This study uses three dataset types: Iris, Wine, and Mushroom, to evaluate the effectiveness of fragmentation method for DDBMSs. Iris is a popular machine learning and statistics dataset, consisting of three classes with fifty instances. Wine Data includes chemical examinations of wines from three varietals in Italy, with 13 components found in different amounts. Mushroom Data includes descriptions of 23 gilled mushroom species, classified as toxic, edible, or unknown edibility. The study aims to compare fragmentation methods for DDBMS..

4.3 Performance analysis

This section analyzes the performance of Vertical and Horizontal fragmentation using EAOA and hybrid AARO algorithms, comparing them with existing algorithms like AOA, CSO, AO, and ARO. It also examines security metrics like encryption and decryption time for the proposed FHE and existing RSA and Blowfish algorithms. Convergence analysis is performed on both models and is shown in figure 1. The proposed EAOA method converges after 50 iterations, compared to the AOA and CSO algorithms which converge after 70 and 90 respectively for vertical fragmentation which is depicted in figure 1(a). The EAOA model obtains the optimal value with a smaller number of iterations due to the advantage of both LFD and OBL in the AOA algorithm, which influences the function of original exploitation and exploration, leading to effective enhancement. The convergence analysis as shown in figure 1(b) for horizontal fragmentation using the hybrid AARO algorithm for the Iris dataset shows the optimal value in 60 iterations, due to the global exploration phase of AO and local exploration phase of ARO. The convergence analysis also shows better results for all vertical fragmentation processes on every dataset. Figures 2 to 4 displays the performance analysis of execution time for Iris, Wine and Mushroom datasets respectively. Execution time is defined as the time taken to complete the clustering process and this analysis is taken for both the vertical and horizontal fragmentation. From the analysis, it is shown that the execution time of EAOA on Iris is 4.8s, Wine is 5.5s and Mushroom 5.4s for vertical fragmentation process. Likewise, the execution time of Iris is 4.8s, Wine is 5.8s and Mushroom is 5.8s for hybrid AARO algorithm to perform the horizontal fragmentation. From the analysis, it is shown that the vertical fragmentation is better than the horizontal fragmentation in execution time analysis. The query processing time analysis is displayed in figure 5 to 7 for each dataset and for both fragmentations. The definition of query processing time is that the time required to process the query for each fragmentation process. The query processing time is different for each dataset as the size of data is different from each other. Also, the result analysis shows that the proposed model of each fragmentation technique displays the smaller processing time compared to the existing algorithms. In addition to this, the vertical fragmentation process is comparatively better than the horizontal fragmentation in query processing time. Figure 8 displays the performance analysis of fragmentation error on each dataset. Here, the fragmentation error is defined as the inefficiency known as unnormalized variance caused by variations from the mean value within a fragment. If the fragment Fi stores the tuples from index $Start\ (Fi)$ to $End\ (Fi)$ and the number of tuples it includes is $Size\ (Fi) = End\ (Fi) - Start\ (Fi)$, then the following equation represents the fragmentation error on both fragmentation approaches.

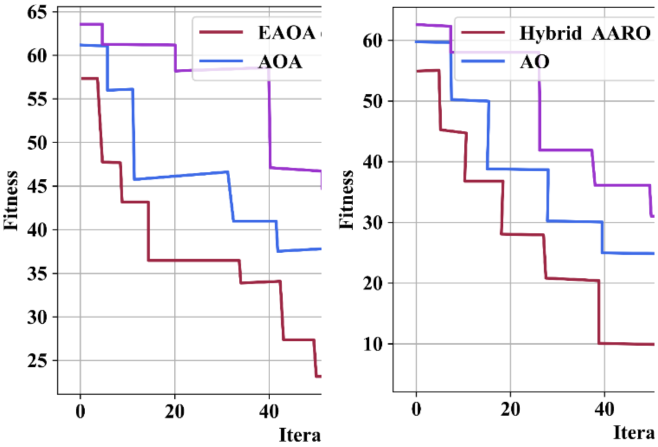
$$E(Fi) = \sum_{x=Start\ (Fi)}^{End\ (Fi)} \left(V(x) - \frac{Value\ (Fi)}{Size\ (Fi)}\right)^2 \qquad (4)$$

From the graph analysis, it is shown that the fragmentation error value is lesser for vertical fragmentation than the horizontal fragmentation approach. Because, the vertical fragmentation allows parallel processing on a relation and increase the clustering efficiency of proposed optimization model due to the frequently used and rarely used attributes are clustered separately.



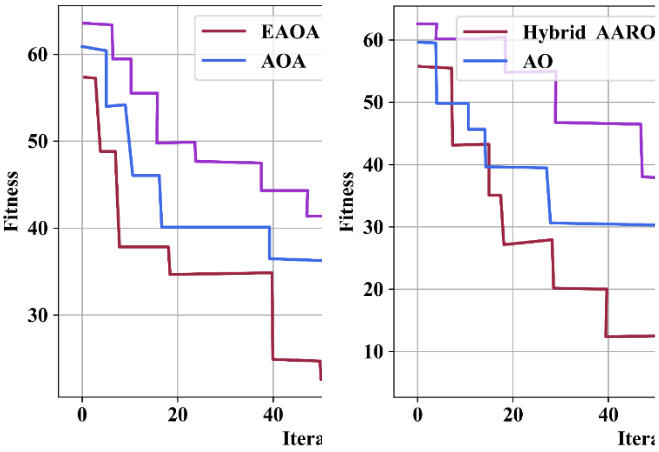
(a) For Vertical fragmentation (Iris)

(b) For Horizontal fragmentation (Iris)



(c) For Vertical fragmentation (Wine)

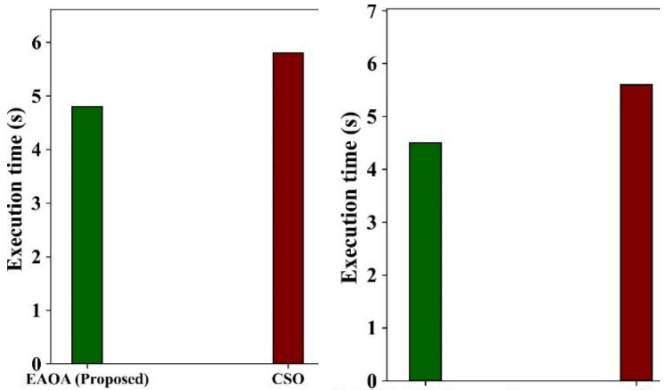
(d) For Horizontal fragmentation (Wine)



(e) For Vertical fragmentation (Mushroom)

(f) For Horizontal fragmentation (Mushroom)

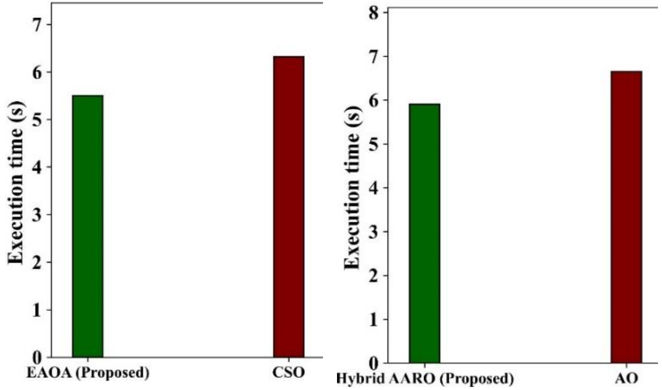
Figure 1: Convergence analysis for all three datasets



(a) For vertical fragmentation

(b) For horizontal fragmentation

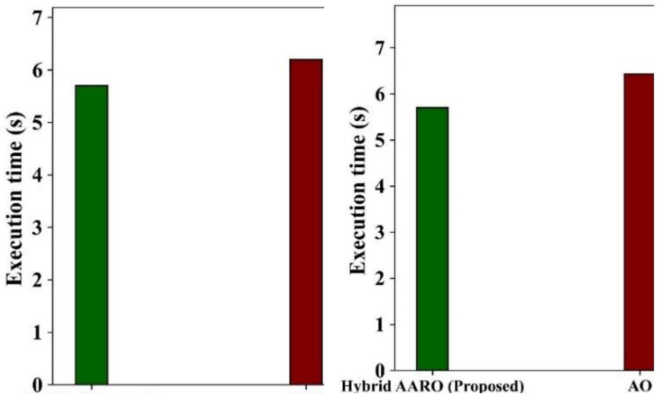
Figure 2: Execution time analysis for Iris dataset



(a) For vertical fragmentation

(b) For horizontal fragmentation

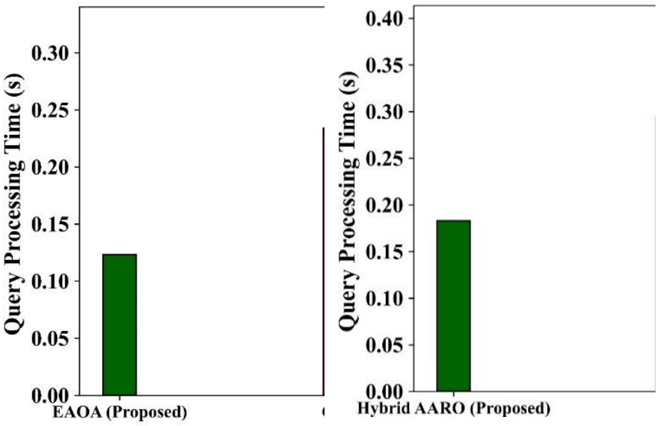
Figure 3: Execution time analysis for Wine dataset



(a) For vertical fragmentation

(b) For horizontal fragmentation

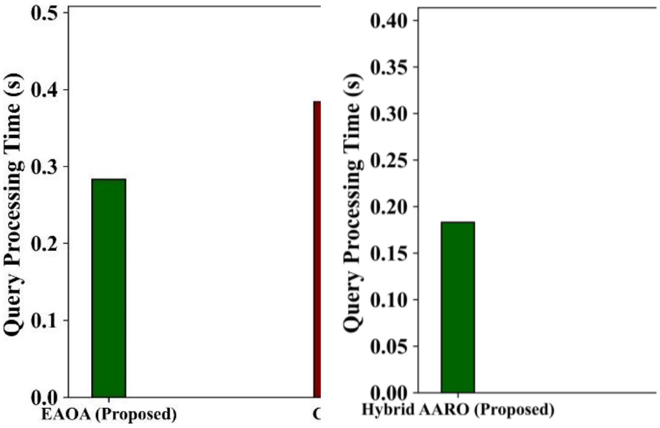
Figure 4: Execution time analysis for Mushroom dataset



(a) For vertical fragmentation

(b) For horizontal fragmentation

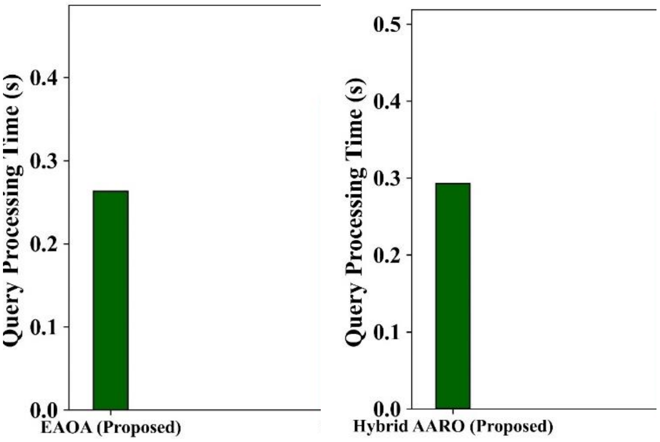
Figure 5: Query processing time analysis for Iris dataset



(a) For vertical fragmentation

(b) For horizontal fragmentation

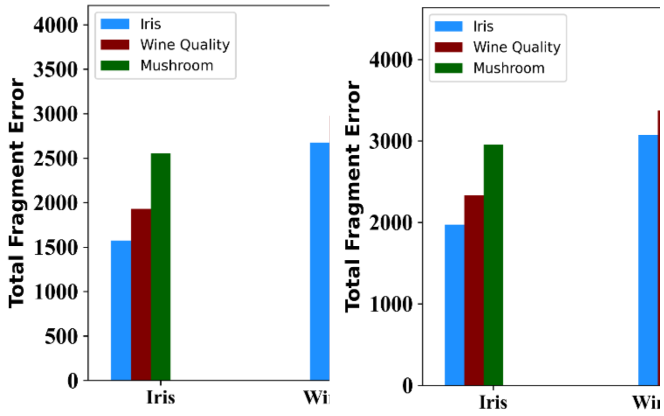
Figure 6: Query processing time analysis for Wine dataset



(a) For vertical fragmentation

(b) For horizontal fragmentation

Figure 7: Query processing time analysis for Mushroom dataset

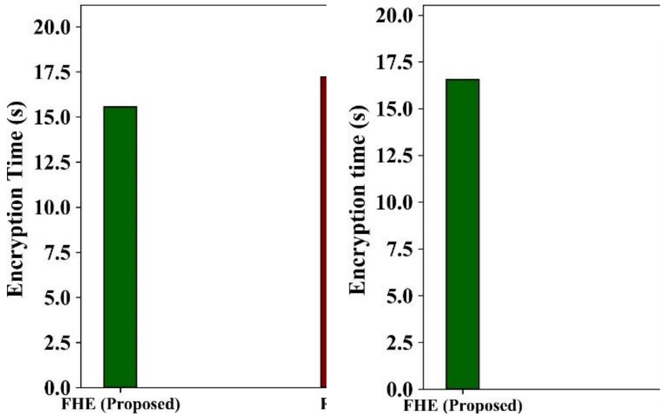


(a) For vertical fragmentation

(b) For horizontal fragmentation

Figure 8: Fragmentation error analysis

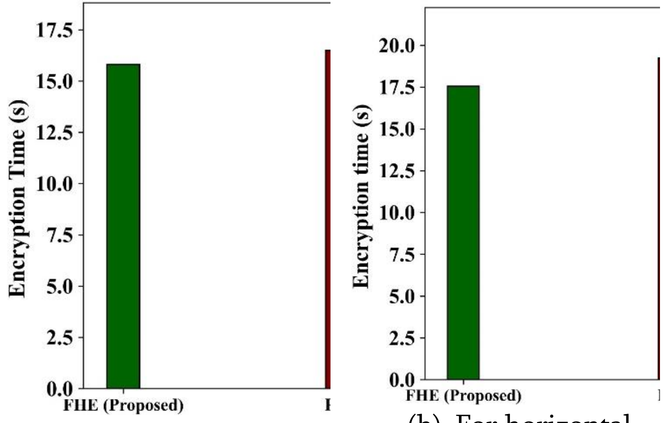
In addition to this, the vertical fragmentation generates the fragments with minimal extraneous data.



(a) For vertical fragmentation

(b) For horizontal fragmentation

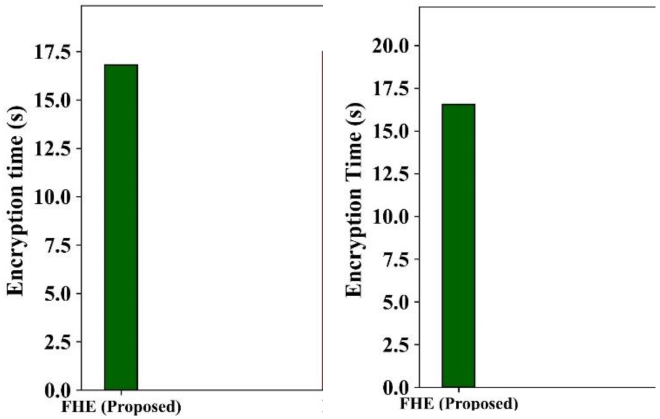
Figure 8: Encryption time analysis for Iris dataset



(a) For vertical fragmentation

(b) For horizontal fragmentation

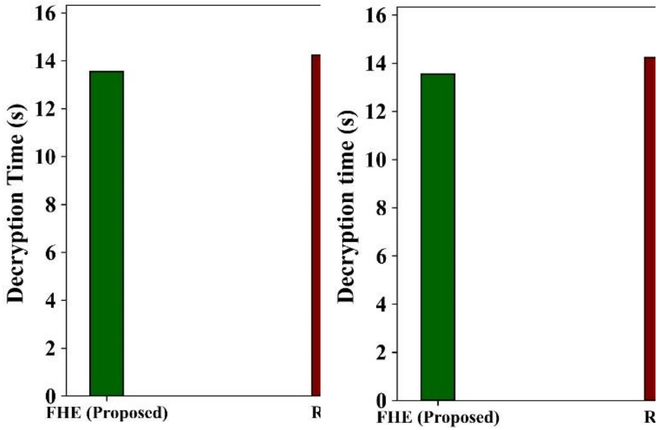
Figure 9: Encryption time analysis for Wine dataset



(a) For vertical fragmentation

(b) For horizontal fragmentation

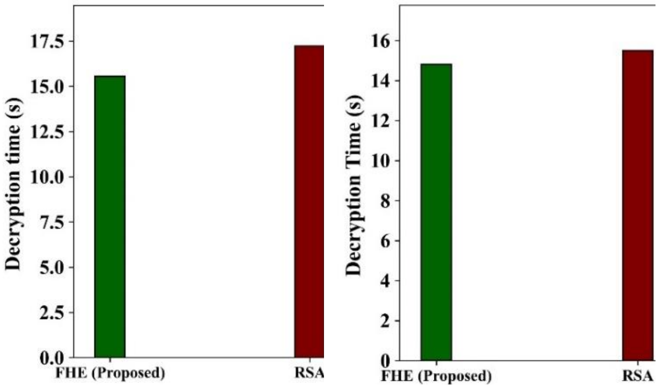
Figure 10: Encryption time analysis for Mushroom dataset



(c) For vertical fragmentation

(d) For horizontal fragmentation

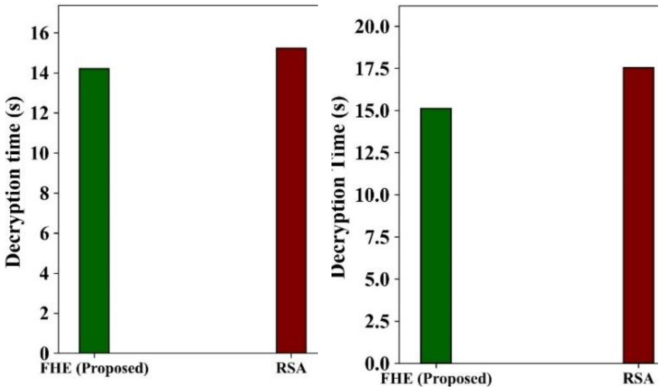
Figure 11: Decryption time analysis for Iris dataset



(c) For vertical fragmentation

(d) For horizontal fragmentation

Figure 12: Decryption time analysis for Wine dataset



(c) For vertical fragmentation

(d) For horizontal fragmentation

Figure 13: Decryption time analysis for Mushroom dataset

The proposed FHE model, compared to existing RSA and AES algorithms, shows lower encryption and decryption times. The analysis of encryption time and decryption time is conducted using standard addition and multiplication operations are shown iin figures 8 to 10 and 11 to 13 respectively. The proposed model's efficiency is evaluated by estimating these parameters. The FHE algorithm is particularly effective in untrusted environments, as it performs both encryption and decryption on the sender and receiver side, resulting in less time spent on both sides. This results in a more efficient and secure security solution.

IV.Conclusion

This research was focused on to design both vertical and horizontal fragmentation in DDBM system using enhanced and hybrid optimization algorithm. For that, EAOA was developed to perform the vertical fragmentation and hybrid AARO algorithm was proposed to perform the horizontal fragmentation to find out the best fragmentation type for DDBM system. Also, a security algorithm namely, FHE was applied to perform the secure data transmission from one to end to another without violating the information presented in the database. Finally, the performance analysis was done for both the fragmentation methods and from that it was proved that the vertical fragmentation is better than the horizonal fragmentation. Because, vertical fragmentation allows tuples to be split so that each part of the tuple is stored where it is most frequently accessed. Also, it clusters the fragmented data with less execution time and retrieves the query response with less processing time. For instance, the execution time of the Mushroom dataset is 5.6s for vertical fragmentation and 5.8s for horizontal fragmentation as well as the query processing time of the Mushroom dataset is 0.26s for vertical fragmentation and 0.29s for horizontal fragmentation. From the result analysis, it was concluded that the vertical fragmentation was better than the horizontal fragmentation for database processing system. In future, developments in database technology will promise to deliver unprecedented scalability, performance, and insights, from the emergence of distributed databases and cloud-based solutions to the incorporation of artificial intelligence and machine learning.

V. REFERENCES

[1]. Nurhayati, Sucinta Tri, and Muhammad Irwan Padli Nasution. "Database Management System Pada Perusahaan." Jurnal Akuntansi Keuangan dan Bisnis 1, no. 2 (2023): 62-64.

[2]. Singh, Rajat, and Pallavi Gildhiyal. "An Innovation Development of Document Management and Security Model for Commercial Database Handling Systems." In 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), pp. 1-6. IEEE, 2023.

[3]. Yang, Xiaofeng. "Improving the Relevance, Speed, and Computational Efficiency of Semantic Search Through Database Indexing: A Review." (2023).

[4]. Manoharan, Siva Ganapathy Subramanian, Rajalakshmi Subramaniam, and Sanjay Mohapatra. "Managing Digital Data–The Biggest Challenge to Today's Organizations." In Enabling Strategic Decision-Making in Organizations Through Dataplex, pp. 27-64. Emerald Publishing Limited, 2023.

[5]. Donner, Eva Katharina. "Research data management systems and the organization of universities and research institutes: A systematic

literature review." Journal of Librarianship and Information Science 55, no. 2 (2023): 261-281.

[6]. Duggineni, Sasidhar. "Impact of Controls on Data Integrity and Information Systems." Science and Technology 13, no. 2 (2023): 29-35.

[7]. Bose, Sudipta, Sajal Kumar Dey, and Swadip Bhattacharjee. "Big data, data analytics and artificial intelligence in accounting: An overview." Handbook of Big Data Research Methods: 0 (2023): 32.

[8]. Punia, Manbir Singh, Kamal Malik, and Vikash Kumar Garg. "Partitioning based multi-persistence model for multi-paradigm database." Measurement: Sensors 25 (2023): 100594.

[9]. Krishan, Kewal, Gaurav Gupta, and Gurjit Singh Bhathal. "A REVIEW AND COMPARISON OF KEY DISTRIBUTED DATABASE CHARACTERISTICS ACROSS SEVERAL NOSQL DISTRIBUTED DATABASES." Journal of Data Acquisition and Processing 38, no. 1 (2023): 321.

[10]. Lehnert, Kerstin, Marthe Klöcking, Dominik Hezel, Lucia Profeta, Peng Ji, and Adrian Sturm. Overcoming Fragmentation of Geochemical Data Resources: Collaboration between EarthChem, Astromat, GEOROC, and MetBase. No. EGU23-16683. Copernicus Meetings, 2023.

[11]. Lovrenčić, Rudolf, and Dejan Škvorc. "Multi-cloud applications: data and code fragmentation for improved security." International Journal of Information Security 22, no. 3 (2023): 713-721.

[12]. Trillo-Montero, David, Samuel Cosano-Lucena, Miguel Gonzalez-Redondo, Juan Jesus Luna-Rodriguez, and Isabel Santiago. "Design and Development of a Relational Database Management System (RDBMS) with Open Source Tools for the Processing of Data Monitored in a Set of Photovoltaic (PV) Plants." Applied Sciences 13, no. 3 (2023): 1357.

[13]. Shepherd, John, Banchong Harangsri, Hwee Ling Chen, and Anne HH Ngu. "A Two-Phase Approach to Data Allocation in Distributed Databases." In DASFAA, pp. 380-387. 1995.

[14]. Shi, Lanying, Kai Wang, Man Li, Jiangang Tong, Hongming Qiao, and Yiquan Jiang. "Research on distributed relational database based on MySQL." In International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2023), vol. 12645, pp. 370-376. SPIE, 2023.

[15]. Mahajan, Shikha, and Leena Jain. "A Study of Distributed Database Management System." International Journal of Engineering and Management Research 13, no. 2 (2023): 121-126.

[16]. Win, Ei Phyu Sin. "LEARNING IN HORIZONTAL SCALING MONGODB DATA SHARDING ON WINDOWS." Journal on Software Engineering 17, no. 3 (2023).

[17]. Vikiru, Allan, Mfadhili Muiruri, and Ismail Ateya. "An Overview on Cloud Distributed Databases for Business Environments." arXiv preprint arXiv:2301.10673 (2023).

[18]. Trillo-Montero, David, Samuel Cosano-Lucena, Miguel Gonzalez-Redondo, Juan Jesus Luna-Rodriguez, and Isabel Santiago. "Design and Development of a Relational Database Management System (RDBMS) with Open Source Tools for the Processing of Data Monitored in a Set of Photovoltaic (PV) Plants." Applied Sciences 13, no. 3 (2023): 1357.

[19]. Gao, Yuan, and Ximu Yan. "Data analysis framework of tourism enterprise human resource management system based on MySQL and fuzzy clustering." International Journal of System Assurance Engineering and Management (2023): 1-13.

[20]. Ascione, Fabio, and Maria Enrica Virgillito. Vertical integration and patterns of divergence in European industries: A long-term input-output analysis. No. 2023/25. Laboratory of Economics and Management (LEM), Sant'Anna School of Advanced Studies, Pisa, Italy, 2023.

[21]. Dhawan, Kamal, John E. Tookey, Ali GhaffarianHoseini, and Mani Poshdar. "Using Transport to Quantify the Impact of Vertical Integration on the Construction Supply Chain: A New Zealand Assessment." Sustainability 15, no. 2 (2023): 1298.

[22]. Huang, Shiyue, Yanzhao Qin, Xinyi Zhang, Yaofeng Tu, Zhongliang Li, and Bin Cui. "Survey on performance optimization for database systems." Science China Information Sciences 66, no. 2 (2023): 121102.

[23]. Mehta, Shikha, Parul Agarwal, Prakhar Shrivastava, and Jharna Barlawala. "Differential bond energy algorithm for optimal vertical fragmentation of distributed databases." Journal of King Saud University-Computer and Information Sciences 34, no. 1 (2022): 1466-1471.

[24]. Alhazmi, Hanan E., Fathy E. Eassa, and Suhelah M. Sandokji. "Towards big data security framework by leveraging fragmentation and blockchain technology." IEEE Access 10 (2022): 10768-10782.

[25]. Noraziah, Ahmad, Ainul Azila Che Fauzi, Sharifah Hafizah Sy Ahmad Ubaidillah, Basem Alkazemi, and Julius Beneoluchi Odili. "BVAGQ-AR for Fragmented Database Replication Management." IEEE Access 9 (2021): 56168-56177.

[26]. Ge, Yong-Feng, Wei-Jie Yu, Jinli Cao, Hua Wang, Zhi-Hui Zhan, Yanchun Zhang, and Jun Zhang. "Distributed memetic algorithm for outsourced database fragmentation." IEEE Transactions on Cybernetics 51, no. 10 (2020): 4808-4821.

[27]. Amer AA, Sewisy AA, Elgendy TM. An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs). Heliyon. 2017 Dec 1;3(12).

[28]. Goli M, Rouhani Rankoohi SM. A new vertical fragmentation algorithm based on ant collective behavior in distributed database systems. Knowledge and Information Systems. 2012 Feb;30:435-55.

[29]. Sewisy AA, Amer AA, Abdalla HI. A novel query-driven clustering-based technique for vertical fragmentation and allocation in distributed database systems. international journal on semantic web and information systems (IJSWIS). 2017 Apr 1;13(2):27-54.

[30]. Sun X, Zhang P, Sookhak M, Yu J, Xie W. Utilizing fully homomorphic encryption to implement secure medical computation in smart cities. Personal and Ubiquitous Computing. 2017 Oct;21:831-9.

Cite this article as :

D. Sahithi, Dr J. Keziya Rani, "A Secure and Optimization Based Clustering for Vertical and Horizontal Fragmentation in Distributed Database Management System", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 6, pp.322-332, November-December-2023. Available at doi : https://doi.org/10.32628/CSEIT2390559
Journal URL : https://ijsrcseit.com/CSEIT2390559