CuLOA-based Data Encryption with Tuned Key for Privacy Preservation in the Cloud

Rajkumar Patil
Department of Computer Science Engineering, GITAM (Deemed to be) University, Hyderabad, Telangana, India
rajkumarpatil45@outlook.com (corresponding author)

Gottumukkala HimaBindu
Department of Computer Science Engineering, GITAM (Deemed to be) University, Hyderabad, Telangana, India
gottumukkalahima@outlook.com

Received: 26 July 2024 | Revised: 28 August 2024 | Accepted: 29 November 2024

 DOI: https://doi.org/10.48084/etasr.8523

ABSTRACT

Preservation of data privacy in cloud computing involves securing sensitive data during analysis and storage. Conventional approaches often use techniques such as encryption and differential privacy, but they can be computationally intensive and may still risk data leakage through indirect inferences. These limitations necessitate advanced methods to balance efficiency and robust privacy. To address this, this study proposes a novel approach using DL-based fine-tuned keys for encrypting the data, aimed at preserving data privacy in the cloud through the Coati Lyrebird Optimization Algorithm (CuLOA) approach. Initially, sensitive data is randomly chosen from the database. Then, the optimal key is derived using the CuLOA approach. This key, along with the sensitive data is input into the SqueezeNet model, which generates a fine-tuned optimal key. Subsequently, the sensitive data are encrypted and stored in cloud storage. Finally, the encrypted data and the optimally tuned key are employed in the data decryption process to recover the original. A comparative experimental analysis showed that the proposed CuLOA approach was better than previous schemes.

Keywords-privacy preservation in the cloud; data encryption; SqueezeNet; data decryption; CuLOA

I. INTRODUCTION

Data plays a crucial role in various domains, providing businesses with valuable location-based insights [1, 2]. Today's world is greatly assisted by the cloud computing industry in several fields, including business, medicine, and education. However, security is a major concern in services. An essential component of a cloud environment is data security [3-5]. Data owners entrust their data to cloud servers for storage, where the data miner extracts valuable information [6]. Utilizing cloud server access reduces substantial data collection expenses, and amalgamating data from numerous transaction sets improves accuracy and reliability. However, cloud servers are not always trustworthy and may inadvertently disclose user data, as evidenced by various cloud leak incidents [7].

Implementing a comprehensive data protection paradigm in complex systems can often be inherently challenging [8]. Privacy preservation in the cloud is gaining increasing attention. Measures include anonymizing sensitive raw data, such as contact details, names, and addresses, within a database to prevent unauthorized access to individuals' personal information. Additionally, sensitive information revealed through data mining processes is also carefully managed to mitigate potential privacy risks [8]. Privacy-preserving association rule mining represents an elegant approach to balancing privacy protection with the benefits of association rule mining. It focuses on protecting the data owner's privacy and the query privacy of the user. Techniques for achieving privacy-preserving association rule mining are categorized into randomization-based and cryptography-based [7]. Several anonymization approaches have been developed to protect privacy when publishing data, including generalization and packetization [9]. However, these approaches are challenging due to high communication and computational costs [10]. To fill this gap, this study introduces a deep learning-based CuLOA approach to encrypt data for privacy preservation in cloud environments.

In [11], an IIoT-based privacy preservation method was developed using AI, consisting of two main stages: data encryption and decryption. In the first stage, unauthorized disclosure was prevented by concealing sensitive IIoT information. This procedure applied the state-of-the-art G–BHO algorithm to generate the best key. Then, a multi-objective function was used, integrating parameters such as degree of modification, information preservation rate, and hiding rate along with the correlation coefficient among

encrypted and decrypted data. This function guided the optimal generation of encryption keys. In [3], a privacy preservation model was presented for cloud environments using AI techniques that can work in business clouds. This approach had high flexibility, agility, and cost savings. This model involved data encryption and decryption and an optimization strategy to select the key. A hybrid algorithm, termed J-SSO, was introduced for this task. Finally, the performance was evaluated compared to conventional methods in terms of different performance measures.

In [10], kubeFlower was presented, which is a privacy-preserving K8s operator. Differential privacy for data management and isolation-by-design were used to address privacy problems. Individual data privacy was protected by differential privacy, while secure resource sharing was ensured by isolation. The Privacy Preserving Persistent Volume Claimer (P3VC) was proposed, a privacy budget-managing technique that augments data with noise. KubeFlower ensures privacy while streamlining FL system maintenance in K8s. This approach was evaluated on a network testbed, made up of several cloud and edge nodes that are geolocated and host FL clients. In [7], MKTFHE was introduced for protecting privacy in association rule mining. First, multi-key homomorphic gates through MKTFHE were developed. Next, a series of computational protocols was designed to ensure privacy using these gates. Finally, a system for privacy-preserving association rule mining was implemented, where a single cloud server served multiple users. Performance evaluation demonstrated the effectiveness and feasibility of this approach.

In [4], the difficulties of directly applying established group-based privacy preservation approaches, namely *l*-diversity and *k*-anonymity, were examined. Formal descriptions of attack approaches were provided and an efficient group-based privacy preservation approach was proposed, which was customized for process mining. This approach incorporated crucial elements such as control-flow patterns, case details, organizational perspectives, and temporal aspects. In addition, flexible and adjustable parameters were recommended to efficiently manage several privacy concerns.

II. PRIVACY PRESERVATION IN THE CLOUD USING THE CULOA APPROACH

This study introduces a deep learning approach for preserving data privacy in cloud environments using the CuLOA approach. Initially, sensitive data is randomly selected from a database. The CuLOA is used to generate the optimal keys, considering constraints such as privacy and the Information Privacy Ratio (IPR). This optimized key and sensitive data are input into the SqueezeNet model to produce an optimally tuned key. The next step involves encrypting the data using the Kronecker product of the optimized tuned key and sensitive data. The encrypted data are then stored in cloud storage. The optimized tuned key and encrypted data are utilized to reconstruct the sensitive data. This procedure is shown visually in Figure 1.

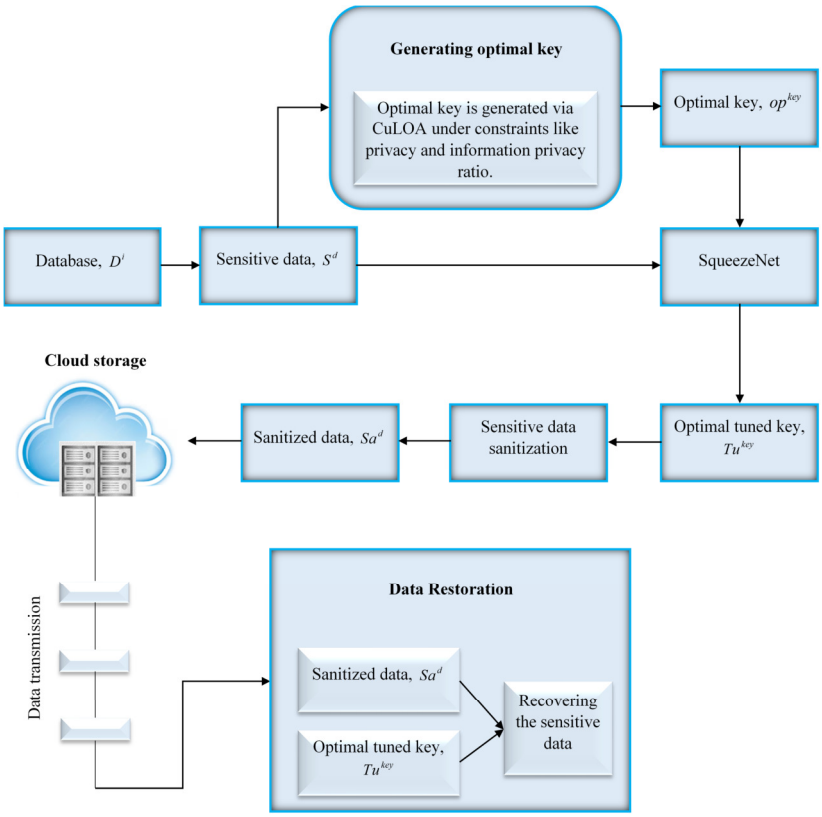

Fig. 1. The overall architecture of the DL-based privacy preservation approach.

A. Data Encryption Process

Consider a database D^i containing data (original data $d^{original}$). Data encryption is also referred to as anonymization or cleansing and plays a critical role in protecting privacy during data mining. Its primary goal is to recognize and eliminate sensitive identifiable information from the database. The data encryption process works to preserve privacy during data mining through the following steps.

1) Extracting Sensitive Data

In the CuLOA approach, sensitive data S^d is randomly selected from the database D^i and processed to ensure privacy preservation. The initial step involves identifying sensitive elements within the dataset D^{set}, such as PII, financial records, health data, or any data that could risk individual privacy. Techniques such as differential privacy ensure that sensitive details cannot be exposed. SMPC and homomorphic encryption further enhance privacy by allowing computations on distributed datasets without revealing individual inputs. These measures collectively ensure that sensitive data S^d is handled with privacy, compliance, and security considerations in mind, protected throughout the extraction process.

2) CuLOA-based Optimization for Optimal Key Generation

After randomly choosing the sensitive data S^d, the next step involves generating an optimal key that adheres to specified constraints for privacy preservation in the cloud. Optimal key generation focuses on selecting or creating keys that enhance utility while minimizing privacy breach risks. This process employs the CuLOA algorithm, which considers two primary constraints: privacy and the information privacy ratio. The CuLOA algorithm effectively identifies keys that balance utility and privacy.

B. CuLOA Algorithm

1) Objective Function

CuLOA is employed to generate the optimal key. This involves the consideration of constraints to maximize both privacy and the IPR. The objective function is represented by:

$$O^F = min\big(w_1 * (1 - privacy) + w_2(1 - IPR)\big) \quad (1)$$

where $w_i = \frac{constraints_i}{sumofconstraints_{(i)}}$ and the respective constraint is explained below.

2) Privacy

Encrypting sensitive private information before processing it on the cloud or any other platform is crucial, as stated in (2):

$$privacy = \frac{1}{d^{original} * S^d} \sum_{i=1}^{d^{original}} \sum_{j=1}^{S^d} \frac{S^d - Sa^d{}_{ij}}{max(S^d - Sa^d{}_{ij})} \quad (2)$$

were Sa^d denotes the length between the original and encrypted data.

3) Information Privacy Ratio (IPR)

IPR is the difference between the number of non-sensitive rules in the encrypted data Sa^d and the number of non-sensitive rules that remain unchanged in them, divided by the total number of non-sensitive rules as expressed in (3):

$$IPR = \frac{Countofnonsensitivedata - Sa^d}{Countofnonsensitivedata} \quad (3)$$

4) Solution Encoding

The solution to the algorithm is a key $a_{i,j}$, according to the following algorithm.

a) Initialization

The CuLOA approach is a metaheuristic algorithm employing lyrebirds as its population [12]. In this iterative process, lyrebirds utilize their searching abilities within the problem-solving space to find optimal solutions. Each lyrebird is indicated as a member of the CuLOA approach, which defines decision variable values based on its location in this space. Mathematically, each lyrebird can be represented by a vector where the respective element corresponds to a decision variable. Together, these lyrebirds constitute the algorithm's population, which can be mathematically modeled through a matrix formulation described in (4). Firstly, the locations of these lyrebirds in the problem-solving space are randomly determined in (5).

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_2 \\ \vdots \\ A_N \end{bmatrix}_{N \times m} = \begin{bmatrix} a_{1,1} \cdots a_{1,j} \cdots a_{1,m} \\ \dots \\ a_{i,1} \cdots a_{i,j} \cdots a_{i,m} \\ \dots \\ a_{N,1} \cdots a_{N,j} \cdots a_{N,m} \end{bmatrix}_{N \times m} \quad (4)$$

$$a_{i,d} = LB_d + ra_{num}.(UB_d - LB_d) \quad (5)$$

In these equations, the population matrix of the CuLOA is specified as A, the i^{th} member of CuLOA is implied as A_i (i.e., candidate solution), the search space of the d^{th} dimension is indicated as $a_{i,d}$ (i.e., decision variable), the numbers of lyrebirds and decision variables are denoted as N and m, ra_{num} is a random number in the interval [0,1], and the d^{th} decision variable of the lower and upper bounds is represented as LB_d and UB_d, respectively. Each lyrebird represents an optimal solution to the problem. The objective function can be evaluated by examining its suggested values for the significant constraints. Thus, the predicted objective function value from CuLOA is represented in (1).

b) CuLOA Mathematical Modeling

In CuLOA, population member locations are iteratively updated based on the lyrebird's decision-making process when it senses danger. This involves two phases: escaping and hiding. Equation (6) is used to simulate the lyrebird's decision to choose between these strategies. Consequently, each CuLOA member's location is updated in the respective iteration based solely on either the escaping or hiding phase.

$$UpdatedProcessForA_i: \begin{cases} BasedOnPhase1, & ra_{num_p} \le 0.5 \\ BasedOnPhase2, & else \end{cases} \quad (6)$$

where ra_{num_p} is a random number in the interval [0,1].

c) Phase 1: Exploration

In this stage, the update of a population member's location in the search space is performed by simulating a lyrebird's

movement from a dangerous spot to a safe habitat. This transition results in significant variations in the lyrebird's location and enables the exploration of various regions in the problem-solving space, demonstrating CuLOA's capability in global search. Also, respective members consider the locations of other population members with a superior value of the objective function in safe habitats. Consequently, the set of safe habitats for respective CuLOA members is defined by:

$$SH_i = \left\{A_k, O_k^F < O_i^F and k \in \{1,2,\dots,N\}\right\}$$
$$i = 1,2,\dots,N, \quad (7)$$

where SH_i denotes the i^{th} lyrebird for the set of safe habitats, and the matrix A from the row has a superior value of the objective function (i.e.,O_k^F) compared to the i^{th} member of the CuLOA (i.e., $O_k^F < O_i^F$). Then, consider that the lyrebird randomly escapes to one of these safe habitats. Following the modeling of lyrebird movement, a new location is evaluated for the respective CuLOA member through (8). If this new location results in an improved value of the objective function, it replaces the earlier location of the corresponding member as specified in (9).

$$a_{i,j}^{L1} = a_{i,j} + ra_{num_{i,j}}.\left(SSH_{i,j} - I_{i,j}.a_{i,j}\right) \quad (8)$$

$$A_i = \begin{cases} A_i^{L1} & O^{F}{}_i^{L1} \le O_i^F \\ A_i & else \end{cases} \quad (9)$$

where the i^{th} lyrebird of the chosen safe habitant is implied as SSH_i, the j^{th} dimension is indicated as $SSH_{i,j}$, the calculation of the new location is denoted as a_i^{L1}, which is the i^{th} lyrebird based on the proposed CuLOA escaping strategy and its dimension is signified as $a_{i,j}^{L1}$, the objective function value is specified as $O^{F}{}_i^{L1}$, and $I_{i,j}$ is a randomly chosen number as 1 or 2. The conventional equation for the lyrebird's new location results in a less efficient new location. Thus, the updated equation for the lyrebird's new location is expressed by:

$$a_{i,j}^{L1} = a_{i,j} * Co^1 + ra_{num_{i,j}}.\left(SSH_{i,j} - I_{i,j}.a_{i,j}.Co^1\right) \quad (10)$$

where $C' = (2\pi)^{0.5}.e^{\left(-0.5\left(\frac{t}{t_{max}}0\right)0\right)}$, C' is the coefficient of the non-linear adjustment, $O^{F}{}_i^{L1}$ denotes the present iteration number, and t_{max} is the maximum iteration number.

d) Phase 2: Exploitation (The Proposed Concept)

Based on the modeling strategy, the location is updated for the population member in the search space. This process ensures that the algorithm exploits safe habitats while avoiding drastic changes that might lead away from optimal solutions prematurely. By continuously evaluating and updating positions based on local improvements in the objective function, CuLOA effectively balances exploration and exploitation, much like how a lyrebird navigates its habitat to find optimal hiding spots.

In the CuLOA approach, a new location is evaluated for the respective member in CuLOA, which is a suitable close area for hiding as expressed in (11). The new position updates the earlier location of the respective member when it enhances the value of the objective function as in (12):

$$a_{i,j}^{L2} = a_{i,j} + \left(1 - 2ra_{num_{i,j}}\right).\frac{UB_j - LB_j}{t} \quad (11)$$

$$A_i = \begin{cases} A_i^{L2} & O^{F}{}_i^{L2} \le O_i^F \\ A_i & else \end{cases} \quad (12)$$

where, based on the hiding strategy, the i^{th} lyrebird's new location A_i^{L2} is evaluated in the CuLOA approach with its j^{th} dimension denoted as $a_{i,j}^{L2}$, and the value of the objective function is denoted as $O^{F}{}_i^{L2}$.

However, the exploitation phase (11) in conventional LOA lacks efficiency and faces issues with global convergence speed. To address this challenge, a hybrid approach combining lyrebird and coati optimization strategies was adopted. Here, the coati is integrated into the lyrebird optimization, resulting in the CuLOA approach utilizing the CuLOA algorithm's strengths. CuLOA demonstrates clear advantages over the lyrebird by effectively balancing global search exploration and local search exploitation. Equation (13) is derived from the exploration phase of the CuOA algorithm [13].

$$a_{i,j}^{L1} = a_{i,j} + ra_{num}.\left(\delta_j - I.a_{i,j}\right)$$
$$for\ i = 1,2,\dots,\left\lfloor\frac{N}{2}\right\rfloor,\ j = 1,2,\dots m \quad (13)$$

where the term δ_j refers to iguana j.

$$a_{i,j}^{L1} = a_{i,j} + ra_{num}.\delta_j - I.a_{i,j} \quad (14)$$

$$a_{i,j}^{L1} = a_{i,j}(1 - I.ra_{num}) + ra_{num}.\delta_j \quad (15)$$

$$a_{i,j}(1 - I.ra_{num}) = a_{i,j}^{L1} - ra_{num}.\delta_j \quad (16)$$

$$a_{i,j} = \left[\frac{a_{i,j}^{L1} - ra_{num}.\delta_j}{(1 - I.ra_{num})}\right] \quad (17)$$

Substituting (17) in (11) gives:

$$a_{i,j}^{L2} = \left[\frac{a_{i,j}^{L1} - ra_{num}.\delta_j}{(1 - I.ra_{num})}\right] + \left(1 - 2ra_{num_{i,j}}\right).\frac{UB_j - LB_j}{t} \quad (18)$$

$$a_{i,j}^{L2} = \frac{a_{i,j}^{L1}}{(1 - I.ra_{num})} - \frac{ra_{num}.\delta_j}{(1 - I.ra_{num})} + \left(1 - 2ra_{num_{i,j}}\right).\frac{UB_j - LB_j}{t} \quad (19)$$

$$a_{i,j}^{L2} - \frac{a_{i,j}^{L1}}{(1 - I.ra_{num})} = \left(1 - 2ra_{num_{i,j}}\right).\frac{UB_j - LB_j}{t} - \frac{ra_{num}.\delta_j}{(1 - I.ra_{num})} \quad (20)$$

$$a_{i,j}^{L2} - \left(1 - \frac{1}{(1 - I.ra_{num})}\right) = \left[\left(1 - 2ra_{num_{i,j}}.\frac{UB_j - LB_j}{t} - \frac{ra_{num}.\delta_j}{(1 - I.ra_{num})}\right)\right] \quad (21)$$

$$a_{i,j}^{L2} = \left\{\frac{\left[\left(1 - 2ra_{num_{i,j}}\right).\frac{UB_j - LB_j}{t} - \frac{ra_{num}.\delta_j}{(1 - I.ra_{num})}\right]}{\left(1 - \frac{1}{(1 - I.ra_{num})}\right)}\right\} \quad (22)$$

Now, (11) is replaced by (22). Finally, the CuLOA optimization algorithm generates the optimal key (op^{key}).

C. SqueezeNet-based Optimal Tuned Key Process

After the CuLOA optimization algorithm generates the optimal key, the sensitive data S^d and the optimal key op^{key} is given as input for this process. This key tuning process aims to further refine and improve the optimal key to achieve a better balance between privacy protection and utilization of the data. This refinement involves fine-tuning the selected key to optimize its performance in preserving privacy while ensuring the effectiveness of data mining tasks.

This study used a SqueezeNet model for the key-tuning procedure. SqueezeNet is recognized for its compact CNN architecture, designed to operate efficiently even in resource-constrained environments [14]. It features "squeeze" layers utilizing 1×1 filters and "expand" layers incorporating a mix of filters such as 1×1 and 3×3, organized into Fire modules. These modules include several convolutional layers, starting with optimal input keys, eight fire modules, and concluding with a convolutional layer**.** Through this process, SqueezeNet iteratively adjusts the parameters of the input optimal key, optimizing its performance to balance privacy protection and data mining effectiveness. The output is an optimally tuned key Tu^{key} that improves privacy preservation while maintaining the utility of the encrypted data for mining tasks.

D. Data Encryption Process

In terms of privacy preservation, data encryption denotes the procedure of removing, obscuring, or anonymizing sensitive information from datasets to protect individuals' privacy. This process is crucial in scenarios where organizations or researchers handle personal data that could identify individuals. By applying the randomly selected sensitive data S^d and the optimally tuned key Tu^{key} with the Kronecker product, the encrypted data Sa^d is attained through as expressed in (23):

$$Sa^d = S^d \otimes Tu^{key} \tag{23}$$

where the symbol $\otimes$ denotes the Kronecker product, which is a mathematical operation on two matrices that results in a block matrix. In terms of privacy preservation in the cloud, the Kronecker product is used to transform sensitive data into an encrypted format. By combining sensitive data with a carefully designed key using the Kronecker product, the data can be obscured, making it difficult for unauthorized users to infer the original information. This transformation helps protect individual privacy while still allowing meaningful data analysis on the encrypted data. Then, the obtained encrypted data Sa^d is stored in the cloud storage.

E. Process of Data Decryption

After the encryption process, the stored encrypted data Sa^d is transmitted into the decryption process for obtaining the sensitive data S^d. In this decryption process, the goal is to recover the original sensitive data S^d from the encrypted data Sa^d. This is achieved through the encrypted data Sa^d and optimally tuned key op^{key} using the Kronecker product as expressed in (24), effectively reversing the encryption process.

$$S^d = Sa^d \otimes op^{key} \tag{24}$$

Initially, the encrypted data transformed to obscure sensitive information is combined with the optimally tuned key op^{key}. The key acts as a critical component in accurately retrieving the original data. The Kronecker product is then applied, which involves a mathematical operation that combines two matrices into a larger matrix in a specific pattern. This operation, when applied to the encrypted data Sa^d and optimally tuned key op^{key}, reconstructs the data back to its original form by leveraging the structural relationships encoded during the encryption process. Therefore, the decryption procedure effectively undoes the encryption, providing access to the sensitive data in its original state.

III. RESULTS AND DISCUSSION

A. Simulation Procedure

The proposed privacy preservation model was implemented using Python 3.7. The simulation was performed on a system with an Intel® Core™ i5-4210U CPU running at 1.70 GHz and 8.00 GB of RAM. The privacy preservation model was evaluated using the UCI Machine Learning Repository Heart disease dataset [15], Dataset 1 (Cleveland), Dataset 2 (Hungary), and Dataset 3 (Switzerland).

B. Analysis of IPR, Privacy, and Fitness

The comparative study on privacy and information privacy ratio analysis examines how the CuLOA approach performs against established methods such as COATI, LBO, KOA, TSO, and BMO, as illustrated in Figure 2. Evaluating privacy preservation ratios is crucial in determining the effectiveness of these techniques in protecting sensitive data. The CuLOA method achieved the highest IPR of 0.298 in Dataset 1, while COATI, LBO, KOA, TSO, and BMO registered minimal privacy ratings. In Dataset 2, the CuLOA method also achieved the highest IPR of 0.895.

Figure 2(c) illustrates a comparative analysis of fitness functions for privacy preservation in both CuLOA and conventional models. In Dataset 3, the CuLOA scheme demonstrated the lowest fitness value of 0.462. In comparison, COATI scored 0.482, LBO achieved 0.527, KOA reached 0.518, TSO attained 0.549, and BMO recorded 0.536.

C. Analysis of Encryption and Decryption

Figure 3 presents the encryption and decryption analysis of the CuLOA method is presented alongside comparisons with COATI, LBO, KOA, TSO, and BMO. Examining the encryption values across the three datasets, a comparative analysis reveals distinct performance among different privacy preservation methods. The proposed CuLOA method consistently yielded the lowest encryption metrics, with values of 0.154, 0.137, and 0.127 observed for Dataset 1, Dataset 2, and Dataset 3, respectively.

When evaluating the effectiveness of decryption on various datasets, the performance of different methods shows significant differences. The CuLOA method consistently achieved high decryption values, with 0.905, 0.910, and 0.929 for Dataset 1, Dataset 2, and Dataset 3, respectively. This indicates its capability to effectively recover and restore original data integrity after privacy-preserving transformations.

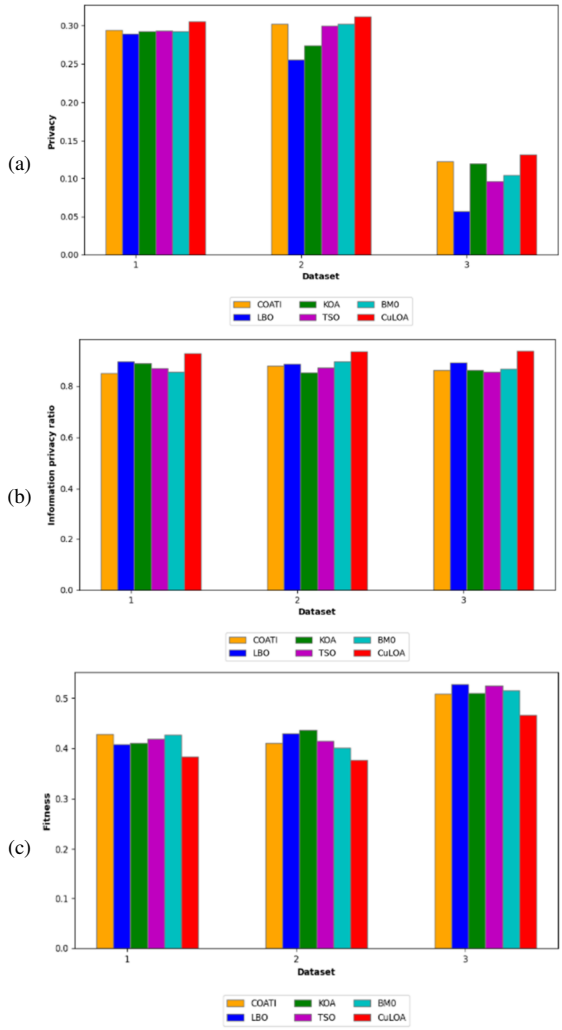

Fig. 2. Comparative analysis of CuLOA and conventional methods: (a) Privacy (b) IPR, (c) Fitness.

IV. CONCLUSION

This study proposed a DL-based privacy preservation approach for data mining using the CuLOA framework. An effective procedure was followed, encompassing data encryption and decryption. First, sensitive data is randomly selected from the database. The CuLOA approach then generates an optimal key, which, along with the sensitive data, is input into the SqueezeNet model to produce an optimal tuned key. Subsequently, the sensitive data are encrypted and stored in the cloud. The stored data are then transmitted to the decryption process to recover the input-sensitive data using the optimally tuned key. A comprehensive analysis, including both simulations and experimental assessments, was performed to evaluate the proposed CuLOA approach, which achieved high decryption values of 0.905, 0.910, and 0.929 for Dataset 1, Dataset 2, and Dataset 3 of the UHI ML repository, respectively.

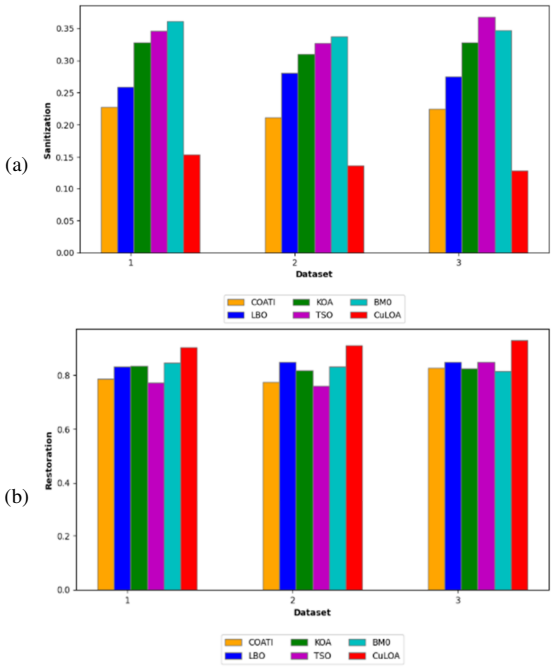

Fig. 3. Comparative analysis of CuLOA and conventional methods: (a) Encryption, (b) Decryption.

REFERENCES

[1] J. S. Lee and S. P. Jun, "Privacy-preserving data mining for open government data from heterogeneous sources," *Government Information Quarterly*, vol. 38, no. 1, Jan. 2021, Art. no. 101544, https://doi.org/10.1016/j.giq.2020.101544.

[2] K. Dave and C. Chand, "Privacy-Preserving in Data Mining using Anonymity Algorithm for Relational Data," *International Journal of Science and Research (IJSR)*, vol. 5, no. 3, pp. 1694–1698, Mar. 2016, https://doi.org/10.21275/v5i3.NOV162221.

[3] D. Ahamad, S. Alam Hameed, and M. Akhtar, "A multi-objective privacy preservation model for cloud security using hybrid Jaya-based shark smell optimization," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part A, pp. 2343–2358, Jun. 2022, https://doi.org/10.1016/j.jksuci.2020.10.015.

[4] M. Rafiei and W. M. P. van der Aalst, "Group-based privacy preservation techniques for process mining," *Data & Knowledge Engineering*, vol. 134, Jul. 2021, Art. no. 101908, https://doi.org/10.1016/j.datak.2021.101908.

[5] J. A. I. S. Masood, M. Jeyaselvi, N. Senthamarai, S. Koteswari, M. Sathya, and N. S. K. Chakravarthy, "Privacy preservation in wireless sensor network using energy efficient multipath routing for healthcare data," *Measurement: Sensors*, vol. 29, Oct. 2023, Art. no. 100867, https://doi.org/10.1016/j.measen.2023.100867.

[6] G. Verma, "Blockchain-based privacy preservation framework for healthcare data in cloud environment," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 36, no. 1, pp. 147–160, Jan. 2024, https://doi.org/10.1080/0952813X.2022.2135611.

[7] P. Jia, J. Zhang, B. Zhao, H. Li, and X. Liu, "Privacy-preserving association rule mining via multi-key fully homomorphic encryption," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 641–650, Feb. 2023, https://doi.org/10.1016/j.jksuci.2023.01.007.

[8] R. R. Nikam and R. Shahapurkar, "Data Privacy Preservation and Security Approaches for Sensitive Data in Big Data," in *Advances in Parallel Computing*, M. Rajesh, K. Vengatesan, M. Gnanasekar, Sitharthan.R, A. B. Pawar, P. N. Kalvadekar, and P. Saiprasad, Eds. IOS Press, 2021.

[9] B. Li and K. He, "Local generalization and bucketization technique for personalized privacy preservation," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 1, pp. 393–404, Jan. 2023, https://doi.org/10.1016/j.jksuci.2022.12.008.

[10] J. M. Parra-Ullauri *et al.*, "*kubeFlower*: A privacy-preserving framework for Kubernetes-based federated learning in cloud–edge environments," *Future Generation Computer Systems*, vol. 157, pp. 558–572, Aug. 2024, https://doi.org/10.1016/j.future.2024.03.041.

[11] M. Kumar *et al.*, "A smart privacy preserving framework for industrial IoT using hybrid meta-heuristic algorithm," *Scientific Reports*, vol. 13, no. 1, Apr. 2023, Art. no. 5372, https://doi.org/10.1038/s41598-023-32098-2.

[12] M. Dehghani, G. Bektemyssova, Z. Montazeri, G. Shaikemelev, O. P. Malik, and G. Dhiman, "Lyrebird Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Biomimetics*, vol. 8, no. 6, Oct. 2023, Art. no. 507, https://doi.org/10.3390/biomimetics8060507.

[13] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, Jan. 2023, Art. no. 110011, https://doi.org/10.1016/j.knosys.2022.110011.

[14] L. S. Bernardo, R. Damaševičius, S. H. Ling, V. H. C. de Albuquerque, and J. M. R. S. Tavares, "Modified SqueezeNet Architecture for Parkinson's Disease Detection Based on Keypress Data," *Biomedicines*, vol. 10, no. 11, Nov. 2022, Art. no. 2746, https://doi.org/10.3390/biomedicines10112746.

[15] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart Disease." UCI Machine Learning Repository, 1989, https://doi.org/10.24432/C52P4X.