

Indian journal of Engineering

To Cite:

Anvar Shathik J, Vishnu Sakthi D, Naveen G, Mosses A, Ramshankar N, Siva Subramanian R. An Intelligent and Privacy-Preserving Cloud Data Storage Framework Using CLCN-DOA Learning and Efficient Quantum Homomorphic Encryption. *Indian Journal of Engineering*, 2026, 23, e15ije1735

doi: <https://doi.org/10.54905/disssi.v23i60.e15ije1735>

Author Affiliation:

¹Dept of Computer Science & Engineering, Anjuman Institute of Technology and Management, Belalkanda, Bhatkal, Karnataka, 581320, India, Email: anvarshathik@gmail.com

²Department of Artificial Intelligence and Data Science, Easwari Engineering College, Ramapuram, Chennai, India, Email: sakthideva@gmail.com

³Dept. of ECE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, SIMATS University, Kancheepuram, Chennai-602105, India, Email: naveeng.sse@saveetha.com

⁴Saveetha School of Engineering, Professor, Department of ECE, Saveetha Institute of Medical and Technical Sciences, Chennai, India, Email: mossesa.sse@saveetha.com

⁵Department of CSE, Saveetha Engineering College, Chennai, Tamilnadu, India, Email: ramshankarn@saveetha.ac.in

⁶Dept of CSE, SRM Institute of Science and Technology, Chennai, India, Email: Sivamr8@gmail.com

Peer-Review History

Received: 18 January 2026

Reviewed & Revised: 12/February/2026 to 29/August/2026

Accepted: 12 September 2026

Published: 30 September 2026

Peer-Review Model

External peer-review was done through double-blind method.

Indian Journal of Engineering
pISSN 2319-7757; eISSN 2319-7765



© The Author(s) 2026. Open Access. This article is licensed under a [Creative Commons Attribution License 4.0 \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

An Intelligent and Privacy-Preserving Cloud Data Storage Framework Using CLCN-DOA Learning and Efficient Quantum Homomorphic Encryption

Anvar Shathik J¹, Vishnu Sakthi D², Naveen G³, Mosses A⁴, Ramshankar N⁵, Siva Subramanian R⁶

ABSTRACT

Cloud data storage is an essential requirement under the framework of the contemporary computing environment, particularly with regard to the growing amount of sensitive data and cyber-attacks. Nevertheless, it is currently hard to guarantee strong anomaly detection and the ability to keep the data confidential because of the complex patterns of user behavior, dynamic contexts of access, and the necessity to conduct privacy-preserving computation. To address these challenges, an advanced framework is proposed, Clifford-Latent Convolutional Network with Dhole Optimization Algorithm (CLCN-DOA) facilitated by Efficient Quantum Homomorphic Encryption (EQHE) for secure cloud data management and anomaly detection. First, pre-processing is performed on the access logs in the cloud which makes use of Adaptive Sliding Window Normalization (ASWN) to remove noise from the records and normalize the behavioral features. A Convolutional Swin-Transformer Network (CSTN) is then used to extract the hierarchical and robust features. The CLCN, in turn, combines Clifford-Steerable Convolutional Neural Networks (CSCNN) and a Latent Attention Network (LAN), leveraging the strengths of both to extract both geometric and behavioral patterns (latent) from the data. The DOA optimizes network parameters to reduce the amount of false positives and improve detection accuracy, and EQHE reduces data confidentiality when stored and processed. Extensive analyses demonstrate that the suggested framework obtains a high accuracy of 99.7%, low-latency of 68 ms, and low-storage overhead of 28.6%. These results demonstrate that the recommended system is strong, effective, and capable of safely storing cloud data and detecting anomalies in real-life settings.

Keywords: Cloud Data Storage, Clifford-Latent Convolutional Network, Dhole Optimization Algorithm, Efficient Quantum Homomorphic Encryption, Convolutional Swin-Transformer Network.

1. INTRODUCTION

With its scalability, flexibility, and cost-effectiveness (Chauhan and Shiaeles, 2023), cloud computing (CC) is another data storage and management paradigm that is recognized and embraced. The use of cloud services has been expanding at a very fast rate, and this has resulted in an increased storage of sensitive data in the cloud environment, like personal data, healthcare records, and business-sensitive files (Abbas et al., 2024). Nevertheless, the benefits come at the cost of major security issues that are posed by cloud data storage because it has a distributed and shared infrastructure. Unauthorized access, data breach, insider threats, and vulnerabilities of services are some of the issues that present great threats to data integrity and secrecy (Krishnan et al., 2023). Because cloud system are accessed via public networks, it is still a significant challenge for researchers and service providers to safeguard the stored data against external and internal attacks (Mostafa et al., 2023).

Encryption is also an important tool in cloud data security because it encodes sensitive data and stores the encrypted form before saving or transmitting the data (Shakor et al., 2024; Devi et al., 2025). Symmetric and public-key encryption are traditional cryptographic tools usually employed in order to guarantee data confidentiality and to stop future unauthorized disclosure (Ahmad et al., 2024). Nevertheless, encryption will not be adequate to curb all the issues of cloud security, especially those that are associated with dynamism in user activity and advanced attack techniques (Guan et al., 2024). Complexity in key management, computational overhead, and low scalability are some of the challenges that diminish the performance of independent encryption mechanisms in a large-scale cloud system (Saxena and Alam, 2023). Moreover, the static security models cannot respond to the changing threats, thus they cannot be used against the current cyber-attacks, which use behavioural and contextual vulnerabilities (Walid et al., 2024).

Deep learning (DL) methods have been widely discussed in recent years as a way of improving cloud safety by means of intelligent data processing and automated decision-making (Man et al., 2023). Learners can process vast amounts of cloud data, access logs, and network traffic to identify unusual patterns and a possible security threat using DL models (Thabit et al., 2023; Krishnamoorthy et al., 2026). The models are more effective in detection than the traditional rule-based approaches and also have lower false alarm rates (Attou et al., 2023). The combination of DL with encryption-based security mechanisms allows the layered defence approach, which entails intelligent threat detection and a high level of data protection (Butt et al., 2023). These types of hybrid secure cloud data storage architectures enhance access control, Data confidentiality, and overall system resiliency, which have the effect of increasing confidence in CC services (Singh et al., 2023).

This work's primary contributions include:

- For a robust framework of secure cloud data storage, a comprehensive framework, namely, CLCN-DOA with EQHE is proposed for anomaly detection and privacy-preserving framework for data management.
- ASWN is introduced to pre-process cloud access logs, reducing noise and normalizing behavioral features while adapting to dynamic access patterns.
- A CSTN is employed to extract hierarchical and robust behavioral features, capturing login patterns, file access sequences, IP/location correlations, and temporal activity trends.
- CLCN is developed by integrating CSCNN with a LAN to effectively detect geometric and latent anomalies in user access behavior.
- The DOA is applied to optimally tune CLCN parameters, improving anomaly detection accuracy, reducing false positives, and enhancing overall model stability.
- Moreover, EQHE provides secure cloud data storage and computation, while preserving high level of data confidentiality, without impacting anomaly detection performance.
- Through comprehensive evaluations, it is proven that the proposed framework can achieve optimal performance, outstanding accuracy, and low latency loss while storing less data than current encryption and anomaly detection mechanisms.
- The framework offers a reliable, expandable, and privacy-preserving solution for secure cloud environments, combining efficient anomaly detection with advanced cryptographic protection.

The remainder of this paper is organized as follows. The existing literature related to cloud data security, encryption, and data deduplication techniques is reviewed. The proposed research methodology and system architecture are then presented in detail. Subsequently, the experimental setup and results are discussed and evaluated. Finally, the key findings and conclusions are presented, along with recommendations and directions for future research.

2. LITERATURE REVIEW

Preethi et al., (2024) designed an intrusion detection system based on DL and encryption, using an encrypted cloud framework. In this study, the distinction between normal and abnormal access behaviour was done using a multiscale deep bidirectional gated recurrent neural network (MDBGRNN) after the structured data had been preprocessed. Cloud data were encrypted using the best encryption system that is the elliptic curve cryptography encryption system in combination with Sine Cosine Egret Swarm Optimization Algorithm (ECC-SCESOA). The encrypted information was also concealed in the steganographic method during storage. The general architecture was to do with intrusion detection, safe encryption and secure cloud storage on one platform.

Ruba and Kalpana, (2025) created a Chunk-based deduplication of data model to enhance security and storage effectiveness in cloud settings. The purpose of the experiment was to establish the redundant information in uploading files, by comparing the file chunks to each other. Data was divided into sections, coded and examined to identify that they were duplicated or not, and stored. To secure non-duplicate data, an adaptive elliptic curve based encryption scheme (AECC) was optimized key-generation. The deduplication task was performed at the client-level, which reduced redundancy at the storage level, maintained data confidentiality and utilized the cloud adequately.

Bharot et al., (2025) developed an encryption-based system named CloudLock to provide safe and reachable data storage throughout multi-cloud systems. The paper has resolved issues that concern the security of data and sharing keys on distributed clouds. An enhanced ChaCha20-Poly1305 encryption scheme was used to secure the stored data, whereas a secure data key exchange scheme was used to provide controlled access to data by authorized users. The framework provided safe data sharing and handling across the various cloud services. Security check and system analysis established that the framework was effective to ensure that confidentiality, accessibility, and operational efficiency were balanced in multi-cloud storage systems.

Shivaramakrishna and Nagaratna, (2023) developed a hybrid cryptographic architecture, which enhances data security in the clouds. The paper has adopted both symmetric encryption and asymmetric encryption (such as RSA) to ensure that the data is credible and confidential. A key management structure has been added to accommodate adaptive approach. An adaptive structured approach has been adopted for facilitating key generation, distribution and final key updating process. Access control that is time-bounded was also implemented to ensure unauthorized access to data. The model emphasized on gaining security of the sensitive cloud data as it was highly resilient in case of unauthorized access in cloud storage system using layered encryption and smart access control.

Gautam et al., (2024) developed an auditing system based on blockchain that will help to ensure the security and trust of cloud data. Lattice-based cryptography was combined with an identity based cryptosystem in the study to deal with the certificate management issue and oppose new attack models. The activities of cloud service providers were monitored with the help of blockchain technology, and there was no need to resort to third-party auditors. Security analysis was proven to be sound on standard cryptographic assumptions. The framework minimised overhead and maximised transparency, reliability, and accountability in cloud auditing, thus enhancing secure data storage and verification procedures.

Rajkumar et al., (2024) developed an architecture of a secure cloud-based data management system based on dynamic chunking and deduplication based on fuzzy logic. The files were broken up into chunks of partially variable size and verified through fingerprinting and hashing. An assignment that effectively organized the information regarding the chunks with the help of an indexing mechanism. A fuzzy logic model that has been optimized by using Ant Lion Optimization (ALO) was used to spot duplicate data. Non-duplicate data was encrypted using Triple Data Encryption Standard (TDES) before being sent to cloud storage. Prior to storage in the cloud, data was encrypted using Triple Data Encryption Standard (TDES) for non-duplicate data. The "frame work" had to aim for reducing the storage overheads and making efficient and secure cloud data management possible.

Suganya and Sasipraba, (2023) developed a secure cloud access control using a combination of Stochastic Gradient Descent, long short-term memory (SGD-LSTM) with Blow Fish encryption (BFE) to identify and block rogue access to the cloud. The system was organized into intrusion prevention, intrusion detection, and user authentication phases. The behavior of the access was analyzed by a DL classifier, and unauthorized users were identified. Encryption was also done to provide secure access to cloud data to legitimate users. The combined solution boosted the security of the cloud by blocking malicious usage and provided better security to the data to be accessed by the user during the process of user authentication and data access.

Guan et al., (2024) developed a safe big data storage model in the cloud environment to eliminate the shortcomings of individual encryption systems. The study had the advantage of enhancing the reliability of metadata by spreading the storage services to several nodes and coordinating them using a distributed management mechanism. Standard data was encrypted using enhanced elliptic curve encryption, and homomorphic encryption was used to compute secure data. One of the strategies was a dual-thread encryption (DTE)

strategy, which made processing faster. The architecture enhanced the ability of fault tolerance, encryption effectiveness, and metadata stability of massive cloud storage systems. As shown in Table 1, the methods adopted in past research studies are summarized.

Alternative techniques have been applied in current studies on secure cloud data storage that have used a combination of intrusion detection, encryption, deduplication and access control. Models structured on DL, including MDBGRNN and LSTM, have been applied to detect abnormal access behavior, and Optimisation techniques have been used to increase the accuracy of detection and the efficiency of encryption. A few works are on minimizing storage overhead by means of chunk-based deduplication, and fuzzy logic. Symmetric-asymmetric hybrids and lightweight and lattice-based cryptography are just some of the cryptographic solutions, as well as cryptographic network structures that combine blockchain with auditing and trust. Though these methods are effective in enhancing security and efficiency, there exist issues involved in key management, scalability, and computational overhead. CC is inexpensive and scalable in data storage, but it suffers from severe security problems, which include unauthorized access, leaked data, and invasion of privacy. The available cloud security systems usually use fixed access control and traditional encryption systems, which cannot identify dynamic and advanced attack patterns. A high number of solutions consider anomaly detection and data encryption as distinct, which results in more false alarms, ineffective key management, and consequent high costs in terms of computational load. Moreover, the approaches are not effective in keeping up with the changing user behavior in the cloud environments. Therefore, a secure cloud storage architecture that is intelligent and integrated to guarantee proper detection of abnormalities, optimum security performance, and efficient data protection is warranted.

3. METHODOLOGY

The suggested CLCN-DOA with EQHE framework will guarantee safe data storage on the cloud and proper identification of anomalies. The pre-processing of cloud access logs by ASWN is done first to eliminate noise and normalize behaviors. CSTN extracts hierarchical features, including patterns of logins, sequencing patterns of file access, IP/location, and time patterns. CLCN is an anomaly detector that incorporates CSCNN and LAN. DOA is used to optimize network parameters, and EQHE provides data security and privacy-preserving computation. The general CLCN-DOA- EQHE's structure is depicted in Figure 1.

3.1. User Access Request

The User Access request phase involves a request initiated by an authorized user in order to upload or download data in the cloud environment. In the process, access-related information like users, type of request, and times of requests, IP address, device information, and access patterns of files is captured by the system. Such access logs and metadata give a fine-grained account of how users interact with that cloud system. The obtained information is used as the main input to the further preprocessing and analytical steps, allowing the framework to trace the access behavior and help with efficient anomaly detection and protected data management.

3.2. Data Preprocessing using Adaptive Sliding Window Normalization (ASWN)

The preprocessing phase of the data is carried out using ASWN (Papageorgiou and Tjortjis, 2025) to optimize the quality and consistency of acquired cloud access logs and carry out feature extraction. The data held in cloud access can usually be noisy, duplicate, unsteady access periods, and inconsistent user behavior, and this can adversely influence the anomaly detection output. ASWN is used to eliminate insignificant information, regularize the differences in access patterns, normalize both numeric and nominal characteristics, and adapt to changing access behavior. This is an adaptive mechanism that allows the framework to manage normal and abnormal access activities effectively. The adaptive window size of each data instance is calculated according to the activity indicator, and it is defined by the following Equations (1) and (2):

$$\varpi\delta(j) = \varpi\delta_{\eta\alpha} \text{ if } \alpha_j = 0 \quad (1)$$

$$\varpi\delta(j) = \varpi\delta_{\alpha} \text{ if } \alpha_j = -1 \quad (2)$$

Here, $\varpi\delta(j)$ denotes the adaptive window size at index j , $\varpi\delta_{\eta\alpha}$ represents the window size for normal access behavior, and $\varpi\delta_{\alpha}$ corresponds to the window size for abnormal access behavior. The activity indicator α_j determines whether the current access pattern is normal or abnormal, allowing the window size to dynamically adjust to changes in access behavior. Based on the selected window size, the sliding window W is constructed as shown in Equation (3):

$$W = \left\{ y_k \mid \max \left(0, j - \frac{\varpi \delta(j)}{2} \right) \leq k \leq \min \left(\eta, j + \frac{\varpi \delta(j)}{2} \right) \right\} \quad (3)$$

In this equation, y_k denotes the access log sample within the window, j and k represent data indices, and η is the sum of the log samples. This adaptive window formulation can be used to make sure that the normalization uses relevant neighboring log entries and avoids distant and irrelevant samples. Lastly, the rescaling of the normalized feature value is done through inverse normalization as shown in Equation (4):

$$y_j = (f^{-1}(z_j, \min_{j-1}, \max_{j-1}) = z_j \times (\max_{j-1} - \min_{j-1}) + \min_{j-1}) \quad (4)$$

Here, y_j represents the normalized value of the feature at index j , while $(\min_{j-1})$ and $(\max_{j-1})$ denote the minimum and maximum feature values obtained from the previous window. This process replaces the feature values with a reasonable scale without compromising temporal and behavioral features. With ASWN preprocessing, noise and spurious log entries are reduced, access behavior trends are regularized, and features are automatically normalized. Consequently, the processed data have become more appropriate for the later stages.

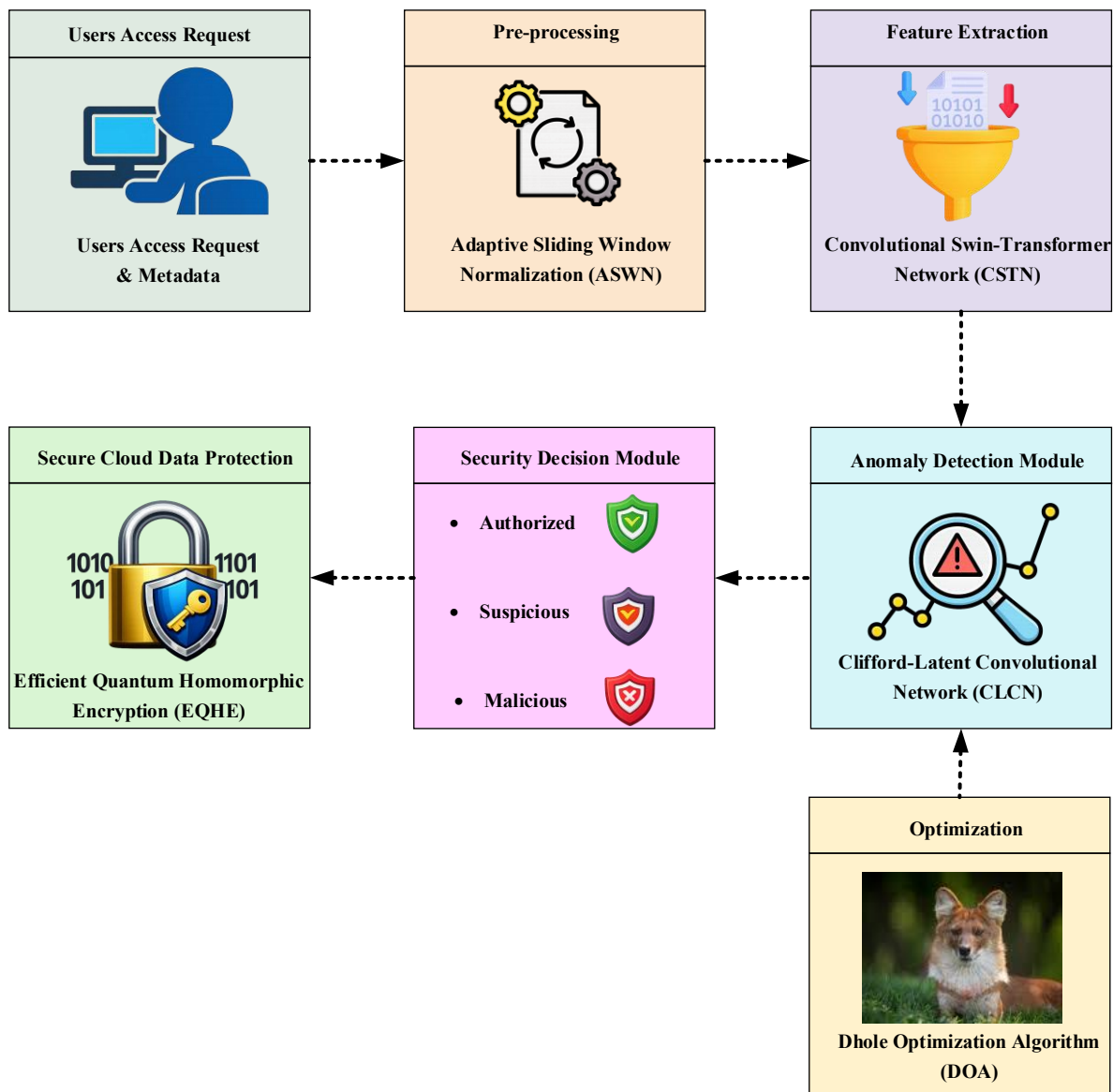


Figure 1. Architecture of the developed CLCN-DOA- EQHE

3.3. Feature extraction using Convolutional Swin-Transformer Network (CSTN)

The normalized cloud access data are subsequently fed to the CSTN (Wang et al., 2024) to isolate the hierarchical and robust aspect of behavior after preprocessing. The CSTN has the motivation and capability of CNNs to capture local features but adds the modeling of long-range dependencies of Transformers, which allows the model to recognize subtle patterns in user access behavior that can be indicative of anomalies.

The patches are tokenized to each data instance, and local and global interactions are calculated with the help of Convolutional Swin-Transformer Blocks (CSTB) and the Depth-convolution Forward Network (DFN). The following main patterns of user behavior are captured during the process of feature extraction:

- **Login frequency & duration** – number and duration of user log-ins.
- **File access patterns** – Which files are visited and in what order.
- **IP address / location** – geolocation, network origin request.
- **Time-of-day activity** – time forms of normal or abnormal use.

CSSTN hierarchical feature extraction process is mathematically described in Equation (5) and (6):

$$\begin{aligned}\hat{a}^m &= wc - msa(convproj(\ln(a^{m-1}))) + a^{m-1} \\ a^l &= dfn(\hat{a}^m) + \hat{a}^m \\ \hat{a}^{m+1} &= swc - msa(convproj(\ln(a^m))) + a^m \\ a^{m+1} &= dfn(a^{m+1}) + \hat{a}^{m+1}\end{aligned}\quad (5)$$

Here, a^l and a^{m+1} denote the output feature embeddings of the m^{th} -th and $m + 1$ -th CSTBs. $\hat{a}^m$ and $\hat{a}^m$ represent intermediate outputs after multi-head self-attention with convolutional projections. $wc - msa$ and $swc - msa$ refer to the (Shifted) Window-based Convolutional Multi-head Self-Attention modules that capture both local and global behavioral patterns. The convolutional projection giving Query (q), Key (k) and Value (v) matrices to compute attention is $convproj(.)$:

$$[q, k, v] = convproj(a^{m-1}) = flatten(depthconv((winpartition(reshape(a^{m-1})))))) \quad (6)$$

$\ln(.)$ denotes layer normalization for stable training. dfn is the Depth-convolution Forward Network, which enhances information interaction across attention windows and preserves local spatial context.

The CSTN can be stacked with multiple CSTBs, which gives it the ability to capture the finer and hierarchical behavioral patterns, such as patterns of login, patterns of file access, patterns of IP and location association, and patterns of the time pattern. The extracted features are then employed in the anomaly detection step to perform a secure and reliable monitoring of cloud data access.

3.4. Anomaly detection using Clifford-Latent Convolutional Network (CLCN)

Once the features have been extracted, anomaly detection is carried out using CLCN, which is the integration of CSCNN (Zhdanov et al., 2024) and a LAN (Zhang et al., 2024). The CSCNN aspect uses Clifford algebra to effectively model multichannel and geometric interactions in the feature space, which allows strong modeling of intricate patterns in user behavior and system activity. At the same time, the latent network component captures high-level latent representation with an emphasis on subtle deviations from normal behavior with no specific attention mechanisms. Through the integration of these two techniques, CLCN is able to detect anomalous access patterns in a very sensitive and accurate way by isolating authorized, suspicious, and malicious activities.

3.4.1. Clifford-Steerable Convolutional Neural Networks (CSCNN)

The first phase of abnormality detection in the secure cloud data storage system is done through the CSCNN. Preliminary discrimination of normal and abnormal access behaviors of users is performed using this network. CSCNN is a neural network based on Clifford algebra that can learn multivector-valued representations of features and thus retain geometric and structural correlations among behavioral features, unlike a conventional convolutional neural network. The goal of this step is to be able to learn small and transformation-inconsistent representations of features of the extracted user activity data, which are critical in the initial level of detecting anomalies reliably. The input to the CSCNN can be defined as a multichannel multivector field, which is written as follows in Equation (7):

$$g: \mathfrak{R}^{\rho, \sigma} \rightarrow c1(\mathfrak{R}^{\rho, \sigma})^d \quad (7)$$

Here, $\mathfrak{R}^{\rho, \sigma}$ denotes the feature space, $c1(\mathfrak{R}^{\rho, \sigma})$ represents the corresponding Clifford algebra, and d indicates the number of feature channels. The representation of heterogeneous access behavior features in this formulation enables the geometric correlations to be maintained. Clifford-steerable convolution is the fundamental functionality of CSCNN, and it can be defined as in Equation (8):

$$M(g_{in})(v) = \int_{\mathfrak{R}^{\rho, \sigma}} L(w)[g_{in}(v - w)]dw, \quad (8)$$

In this equation, g_{in} depicts the input feature map, $L(w)$ denotes the convolution kernel, and v indicates the spatial location. This operation allows CSCNN to obtain local geometric patterns of user access behaviours. The convolution kernels are steerable, i.e., they obey the steerability constraint in Equation (9):

$$L(hw) \stackrel{!}{=} \varsigma_{c1}^{c_{out}}(h)L(w)\varsigma_{c1}^{c_{in}}(h^{-1}) =: \varsigma_{hom} \quad (9)$$

Here, $h \in P(\rho, \sigma)$ represents an orthogonal transformation, while $\varsigma_{c1}^{c_{in}}$ and $\varsigma_{c1}^{c_{out}}$ denote the Clifford representations for the input and output feature spaces, respectively. This limit makes the learned features equivariant to geometric transformation, which enhances the resilience of anomaly detection. CSCNN output provides information that shows the possible deviation of the normal access pattern by identifying structural and spatial anomalies in the feature space. These polished representations serve as the foundation of the first level of anomaly detection and are later sent to the second step of the framework as latent-level analysis and ultimate security decisions formulation.

3.4.2. Latent Attention Network (LAN)

This designates the next-level anomaly-detecting procedure based on the LAN, where the output feature representations acquired by the CSCNN are sent to further behavioral categorization. As CSCNN is dedicated to the acquisition of geometric and structural aberrations at the first tier, LAN is developed as the model of latent associations and situational correlations between user access patterns. This step tries to improve anomaly detection by focusing on the latent relationships amongst access characteristics so that a more correct distinction is made between normal and suspicious activities.

The LAN starts with the creation of latent representations that code contextual relationships of the CSCNN output features. In order to accomplish this, the relative feature importance is modeled as a latent weighting mechanism, formulated as Equation (10):

$$Y_j = \text{soft max}(RB_j X_R \cdot (Q X_Q)^T) Y_j. \quad (10)$$

Here, Y_j denotes the input feature vector from CSCNN, RB_j represents the latent encoding of feature relationships, Q corresponds to contextual embeddings, and X_R and X_Q are learnable projection matrices. This formulation allows LAN to give feature components adaptive weights depending on their latent relevance. LAN ensures contextual continuity over feature sequences by keeping a latent state, which is a summation of past behavioral information. The reset gate update in Equation (11) is as follows:

$$s_u = \varsigma(Y_u X_{Ys} + D_{u-1} X_{Ds}) \quad (11)$$

where Y_u is the current input feature vector, D_{u-1} denotes the previous latent state, X_{Ys} and X_{Ds} are learnable parameters, and $\varsigma(\cdot)$ represents the sigmoid activation function. Equally, the update gate that regulates the effect of the past latent information is determined in Equation (12):

$$v_u = \varsigma(Y_u X_{Yv} + D_{u-1} X_{Dv}) \quad (12)$$

This gate controls the amount of historical behavioral context to be remembered in terms of assessing current anomalies. The candidate latent state is calculated by the following equation as presented in Equation (13):

$$\tilde{D}_u = v(Y_u X_{vi} + (s_u \circ D_{u-1}) X_{Di}) \quad (13)$$

where $v(\cdot)$ denotes the hyperbolic tangent activation function, s_u is the reset gate, D_{u-1} is the previous latent state, and $\circ$ represents multiplication by elements. The last latent state update is done with the help of Equation (14):

$$D_u = v_u \circ D_{u-1} + (1 - v_u) \circ \tilde{D}_u \quad (14)$$

This update algorithm guarantees the steady propagation of contextual information when working on feature chains, so that LAN can learn long-term dependencies in the behavior of user access. The refined features are then adaptively combined with the original CSCNN output in order to increase the discrimination ability after latent modeling. This amalgamation is written down as: Equation (15):

$$Y = \text{norm}(Y + g(Y \circ \beta + (1 - \beta) \circ D)) \quad (15)$$

where Y denotes the feature representation, D is the latent state, β represents adaptive fusion weights, $g(\cdot)$ is a nonlinear transformation, and $\text{norm}(\cdot)$ indicates normalization. The combination improves important behavioral patterns and inhibits insignificant variations. Overall, LAN uses next-level anomaly detection methods by modeling latent contextual relationships in features extracted through CSCNN. The smooth representations generated by LAN allow the proper classification of user access activity and are the foundation of the next stage. After this, the DOA optimizes the weight parameters of the CLCN.

3.4.3. Dhole Optimization Algorithm (DOA)

The DOA (Mohammed et al., 2025) is an evolutionary metaheuristic that is based on cooperative hunting, vocalization and adaptive attack behavior of dholes. In this context, DOA is utilized to optimize the X_{Y_S} and X_{D_S} parameters of the CLCN to improve the performance of the CLCN with regard to secure and accurate classification of cloud data and decision making.

Step 1: Population Initialization

DOA starts with a population of candidate solutions with each Dhole corresponding to a potential solution within the search area. The location of every Dhole is associated with a group of decision variables to be optimized. The Dhole population can be mathematically defined as in Equation (16):

$$E = \begin{bmatrix} e_{1,1} & \cdots & e_{1,k} & \cdots & e_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{j,1} & \cdots & e_{j,k} & \cdots & e_{j,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{O,1} & \cdots & e_{O,k} & \cdots & e_{O,n} \end{bmatrix}_{O \times n} \quad (16)$$

where E denotes the Dhole population matrix, e_j represents the j^{th} Dhole (candidate solution), O is the total number of Dholes, n is the number of decision variables, and $e_{j,k}$ denotes the value of the k^{th} decision variable of j^{th} Dhole.

Step 2: Fitness Evaluation

A classification performance of the CLCN is used to evaluate each Dhole (candidate solution). The fitness function is computed as in Equation (17):

$$\text{Fitness function} = \text{Optimize}(X_{Y_S} \text{ and } X_{D_S}) \quad (17)$$

where X_{Y_S} enhances the accuracy and X_{D_S} maximizes the recall.

Step 3: Searching Stage (Exploration Phase)

This phase allows the exploration of the world, making Dhohes move to the good areas of the search space with a diversity that avoids early convergence. This is given in Equations (18-20):

$$prey = (prey_{local} + prey_{global})/2, \quad (18)$$

$$E_{j,k}^{u+1} = e_{j,k}^u + D_2 \times rand \times (prey_k - e_{j,k}^u), \quad (19)$$

$$D_2 = 1 - (u/U), \quad (20)$$

where $prey_{local}$ denotes the most effective option for the existing population, $prey_{global}$ represents the optimal outcome from every iteration, $e_{j,k}^u$ and $E_{j,k}^{u+1}$ denote the current and updated positions of the j^{th} Dhohle in the k^{th} dimension, u is the current iteration, U is the highest number of iterations, D_2 is a decreasing exploration coefficient, and $rand$ is a random number in $[0,1]$.

Step 4: Encircling Stage (Collaborative Exploration)

During this stage, the Dhohes work together in circling the prey by moving when it comes to the other pack members, and this enhances exploration and optimization of a solution. It is given in the following equations (21) and (22):

$$E_{j,k}^{u+1} = e_{j,k}^u - e_{a,k}^u + prey_k, \quad (21)$$

$$a = 1 + round(rand \times (c - 1)), such\ that\ j \neq k, \quad (22)$$

where $E_{j,k}^u$ denotes the position of a randomly selected Dhohle, a is the index of the selected Dhohle, O is the total population size, and $prey_k$ represents the prey position in the k^{th} dimension.

Step 5: Attacking Stage (Exploitation Phase)

This stage heavily exploits the optimal solution by simulating the strength of prey and carrying out adaptive attack mechanisms, which hasten the solution to the optimum. This step takes the form of Equations (23-26):

$$T = D_3 \times rand \times (fitness_j / fitness_{prey}), \quad (23)$$

$$X_{prey} = exp\left(-\frac{1}{T}\right) \times prey_{local}, \quad (24)$$

$$E_{j,k}^{u+1} = e_{j,k}^u + X_{prey} \times qt \times (\cos(2 \times \pi \times rand) - \sin(2 \times \pi \times rand) \times X_{prey} \times qt), \quad (25)$$

$$E_{j,k}^{u+1} = (e_{j,k}^u - prey_{global}) \times qt + qt \times rand \times e_{j,k}^u, \quad (26)$$

where T denotes the prey size, D_3 is the prey scaling factor, $fitness_j$ represents the fitness of the j^{th} Dhohle, $fitness_{prey}$ denotes the prey fitness value, X_{prey} represents the weakened prey strength, qt is the hunting suitability factor, and $prey_{local}$ and $prey_{global}$ represent local and global best solutions, respectively.

Step 6: Termination Criterion

The algorithm continues updating the Dhohle population until the greatest quantity of iterations is met or convergence is reached, and the best Dhohle at the end is the optimized parameter set to the CLCN model. Figure 2 displays the flowchart of DOA.

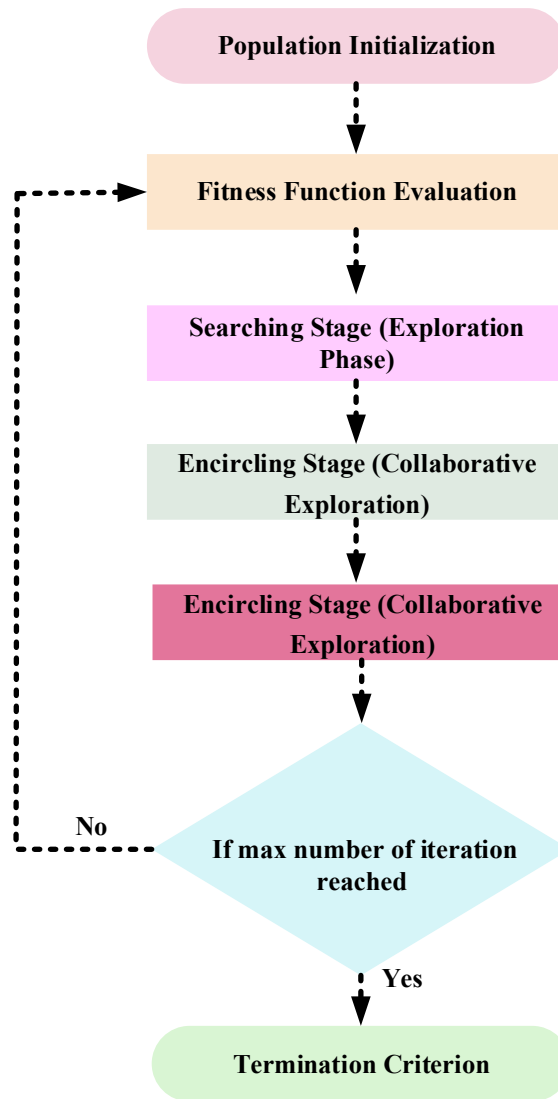


Figure 2. Flowchart of DOA

3.5. Decision Layer: Security Decision Module

In the decision Layer, the anomaly scores generated by the optimized CLCN are converted to explicit security decisions. To facilitate proper security measures, this module categorizes every access request as authorized, suspicious, or malicious. It is given in Equation (27):

$$T_j = \varsigma(g_{CLCN}(y_j)) \quad (27)$$

Let the output of the CLCN for the j^{th} access request be denoted as an anomaly confidence score T_j , obtained after the final activation layer. The score reflects the likelihood of abnormal behavior and is bounded within the interval $[0,1]$. The security state of the access request is calculated with reference to the predefined thresholds of decisions based on Equation (28):

$$E_j = \begin{cases} Authorized, T_j < \tau_1 \\ Suspicious, \tau_1 \leq T_j < \tau_2 \\ Malicious, T_j \geq \tau_2 \end{cases} \quad (28)$$

where E_j denotes the security decision, τ_1 is the authorization threshold, and τ_2 is the malicious activity threshold, with $0 < \tau_1 < \tau_2 < 1$.

In case the access request is defined as Authorized, then normal operations of data access or storage are allowed. Any suspicious requests are under additional verification. Attempts whose tags are marked as Malicious are blocked and logged to undergo further forensic examination.

3.6. Secure Cloud Data Protection using Efficient Quantum Homomorphic Encryption (EQHE)

The use of EQHE (Liu et al., 2022) allows the calculation of quantum data in a secure manner without exposing plaintext information. In the suggested secure cloud data storage model, EQHE will provide confidentiality in the outsourced quantum processing and will permit authorized assessors to collaboratively execute quantum actions.

i) Key Distribution Stage

This step entails the distribution of secret parameters to many quantum evaluators to allow joint computation. This aims at making sure that no individual evaluator can individually decrypt or deduce the original quantum data to ensure distributed trust and access control. The constraint of secret sharing is Equation (29):

$$c_1 y_{t1} + c_2 y_{t2} + \dots + c_e y_{te} = 1 \quad (29)$$

where c_j denotes the selected private coefficients, y_{tj} represents the corresponding solution values, and e is the number of cooperating evaluators. This formula ensures that the process of decryption can only be possible with the collaboration of the evaluators.

ii) Encryption Stage

In encryption, the initial quantum state is rotated using a unitary operator that is based on rotations so that its computational basis is concealed. This will guarantee that cloud evaluators are only working with encrypted quantum states and maintain the confidentiality of data during processing. The quantum state of the encrypted state is manifested in Equation (30):

$$|\varphi_n^0\rangle = V(\theta_0)|\varphi_n^0\rangle \quad (30)$$

where $V(\theta_0)$ denotes the quantum rotation operator, $\theta_0 = 2\pi - \zeta_1 - \zeta_2$ is the encryption angle, ζ_1 and ζ_2 are secret random parameters, and $|\varphi_n^0\rangle$ is the original input state.

iii) Sharing and Evaluation Stage

This stage involves the encryption of quantum states, which are then evaluated sequentially by a series of evaluators. All the evaluators use approved quantum gates and do not access any plaintext data, and the computational leaks of information are blocked with intermediate masking parameters. The transformation of the evaluator is determined in Equation (31):

$$|\varphi_e^n\rangle = H_e^n V((-1^{\kappa_{sum}} \theta_e)) |\varphi_{e-1}^n\rangle, \quad (31)$$

where H_e^n represents the unitary gate applied by the n^{th} evaluator, θ_e is the evaluator-specific rotation parameter, κ_{sum} accumulates gate-dependent parity values, and $|\varphi_e^n\rangle$ is the updated encrypted state.

iv) Decryption Stage

The last operation of decryption restores the correct result of computation upon completion of all the encryption operations. The data owner is able to eliminate the cumulative encryption effects with evaluators using secret correction operators without communicating with them. The process of decryption is carried out in Equation (32):

$$E|\varphi_n^e\rangle = H_n|\varphi_n\rangle \quad (32)$$

where E denotes the decryption operator, $|\varphi_n^e\rangle$ is the final evaluated quantum state, and H_n represents the intended quantum computation. It ensures that this is right and privacy is maintained. The EQHE module makes sure that sensitive cloud-stored data is

encrypted during processing so that it does not leak data to untrusted cloud servers. Distributed authorization, integrity verification, and correctness verification, EQHE is a powerful tool that enhances the integrity and confidentiality of cloud data storage systems.

3.6.1. Quantum Data Representation and Security Considerations

The EQHE component is incorporated into the proposed framework to provide privacy-preserving processing of sensitive cloud-stored data. Before applying the EQHE operations, the conventional cloud data or intermediate data representation is mapped to the quantum-state representation required by the encryption procedure. The resulting quantum state is subsequently processed through the encryption, sharing and evaluation, and decryption stages described in Equations (29)–(32). Thus, the EQHE module is used as a secure processing layer, while the remaining components of the proposed framework perform cloud access analysis, anomaly detection and feature extraction, and optimization.

Encryption is carried out by application of parameterized operators is unitary rotation and random secret parameters. Multiple, cooperating evaluators share the encrypted quantum state and carry out authorized quantum operations, without directly accessing the original plaintext state. Each evaluator does not have access to necessary parts of the key to reconstruct or decrypt all the underlying data. Only in the decryption process the original result is obtained after applying the corresponding bi-phases operations after correction operations.

The security objective of the EQHE module is to protect the confidentiality of the input data during outsourced processing. The security assumptions considered in the proposed model include the presence of untrusted or semi-trusted cloud evaluators that can perform authorized computations on encrypted data but should not obtain direct access to the underlying plaintext information or the complete set of secret parameters. The distributed evaluation mechanism is therefore intended to reduce the risk associated with a single evaluator accessing sufficient information to reconstruct the original data.

The EQHE component in the present study is evaluated within the proposed simulation framework and mathematical formulation described by Equations (29)–(32). The study does not claim execution on physical quantum hardware. Therefore, the results should be interpreted as a simulation-based evaluation of the proposed secure processing mechanism rather than as a hardware-level quantum implementation. Factors like the quantity of actual qubits, hardware-specific gate errors, and device-dependent circuit depth are not evaluated in the present experimental setup and remain important considerations for future implementation on quantum simulators or quantum hardware.

Furthermore, the reported security and confidentiality performance values are used as comparative evaluation measures within the experimental environment and should not be interpreted as standardized cryptographic security levels. The proposed framework aims to demonstrate improved confidentiality during the simulated cloud data-processing process; however, formal security guarantees expressed in terms of established cryptographic notions, adversarial advantage, security parameters, or bit-security levels require a separate formal security analysis. Such an analysis, together with quantum circuit resource evaluation and implementation on quantum simulators or hardware, is identified as future work.

4. RESULTS & DISCUSSION

In this section, the experimental test of the designed EQRDGANDRA framework will be provided with references to whether it is effective in the provision of safe and efficient cloud data storage and analysis.

4.1. Dataset and Simulation Environment

The proposed CLCN-DOA with EQHE was experimentally tested in a simulated cloud environment to test secure cloud storage and anomaly detection. For the simulation, a large cloud infrastructure with 1,000 virtual nodes in various data centers was used. Up to 10 GB of simulated dataset size and a 1 Gbps bandwidth in the cloud environment were considered.

The experimental data were generated in the form of cloud access logs representing user interactions with the cloud environment. During each user access request, the system captured access-related information, including user information, request type, request time, IP address, device information, and file access patterns. These access logs and metadata were used as the input for the subsequent preprocessing, feature extraction, and anomaly detection stages.

The generated access activities were designed to represent variations in user behavior within the simulated cloud environment. The behavioral information included login frequency and duration, file access patterns and sequences, IP address and location information,

and time-of-day activity patterns. These characteristics enabled the proposed framework to analyze both regular and irregular access behavior and identify possible deviations associated with suspicious or anomalous activities.

To minimize noise and normalize behavioral characteristics, we first removed it using Adaptive Sliding Window Normalization (ASWN) of the generated cloud access logs. Later, the normalized access information was given to the Convolutional Swin-Transformer Network (CSTN) for obtaining hierarchical behavioral features. The extracted features were then input to the proposed Clifford d-Latent Convolutional Network (CLCN) for anomaly detection and classification of access requests as either authorized, suspicious, or malicious.

The use of a simulated cloud environment enabled the proposed framework to evaluate cloud access behavior across multiple distributed virtual nodes under controlled conditions. However, the present experimental evaluation is based on simulated or controlled cloud access logs. Therefore, validation using publicly available benchmark datasets and real-world cloud system logs will be considered in future work to further investigate the generalizability and practical suitability of the suggested framework.

4.2. Experimental Setup

The proposed CLCN-DOA model with EQHE were done in Python and run in Windows 10 (64-bit), having Intel Core i7 processor and 16 GB of RAM. The experiments were run in the simulated cloud setup, which involves 1000 virtual nodes spread in the various data centers. The main parameters of the simulation used in the experimental evaluation are given in Table 1.

Table 1. Simulation Parameters

Parameters	Description
Number of virtual nodes	1,000
Dataset type	Simulated cloud access logs
Dataset size	Up to 10 GB
Bandwidth	1 Gbps
Processor	Intel Core i7
RAM	16 GB
Operating system	Windows 10 (64-bit)
Implementation platform	Python

Table 2. Performance Metrics

Detection metrics			
Accuracy	$\frac{\mu\nu+\mu o}{\mu\nu+\mu o+\varpi\nu+\varpi o}$	Precision	$\frac{\mu\nu}{\mu\nu+\varpi\nu}$
Specificity	$\frac{\mu o}{\mu o+\varpi\nu}$	F1-Score	$\frac{2*(Precision*Sensitivity)}{Precision+Sensitivity}$
Recall	$\frac{\mu\nu}{\mu\nu+\varpi o}$		
Secure storage metrics			
Storage Overhead	$\frac{EDS-ODS}{ODS} \times 100$	Upload time (ms)	$T_{up}^{end} - T_{up}^{start}$
Latency (ms)	$T_{res} - T_{req}$	Throughput (MB/s)	$\frac{D}{T}$
Encryption Time (ms)	$T_{enc}^{end} - T_{enc}^{start}$	Decryption Time (ms)	$T_{dec}^{end} - T_{dec}^{start}$
Security Level (%)	$1 - \frac{SA}{TA} \times 100$	Data Confidentiality Rate (%)	$\frac{PDB}{TDB} \times 100$

Where, $\mu\nu$ denotes true positive, $\mu\phi$ denotes true negative, $\phi\nu$ denotes false positive, $\phi\phi$ denotes false negative, EDS represents the encrypted data size, ODS represents the original data size, T_{up}^{start} and T_{up}^{end} indicate the upload start and end times, T_{req} and T_{res} denote the request and response times, D represents the total data transferred, T denotes the total transmission time, T_{enc}^{start} and T_{enc}^{end} indicate the encryption start and end times, T_{dec}^{start} and T_{dec}^{end} indicate the decryption start and end times, SA denotes the number of successful attacks, TA denotes the total number of attack attempts, PDB represents the protected data blocks, and TDB represents the total data blocks.

4.3. Performance metrics

The classification metrics and system-level cloud security metrics are used to measure the performance of CLCN-DOA -EQHE. Detection performance is assessed with the help of Precision, Recall, Specificity, and Accuracy, F1-score, and the efficiency of encryption and cloud storage can be specified with Storage Overhead, Upload Time, Latency, Throughput, Encryption and Decryption Time, Security Level, and Data Confidentiality Rate. Table 2 shows the mathematical formulas of the performance measures adopted.

4.4. Performance analysis

The section will evaluate the performance of the suggested framework in terms of training accuracy, testing accuracy, and loss values that were achieved throughout the learning process. These metrics are employed to determine the effectiveness and capability of generalization of the model.

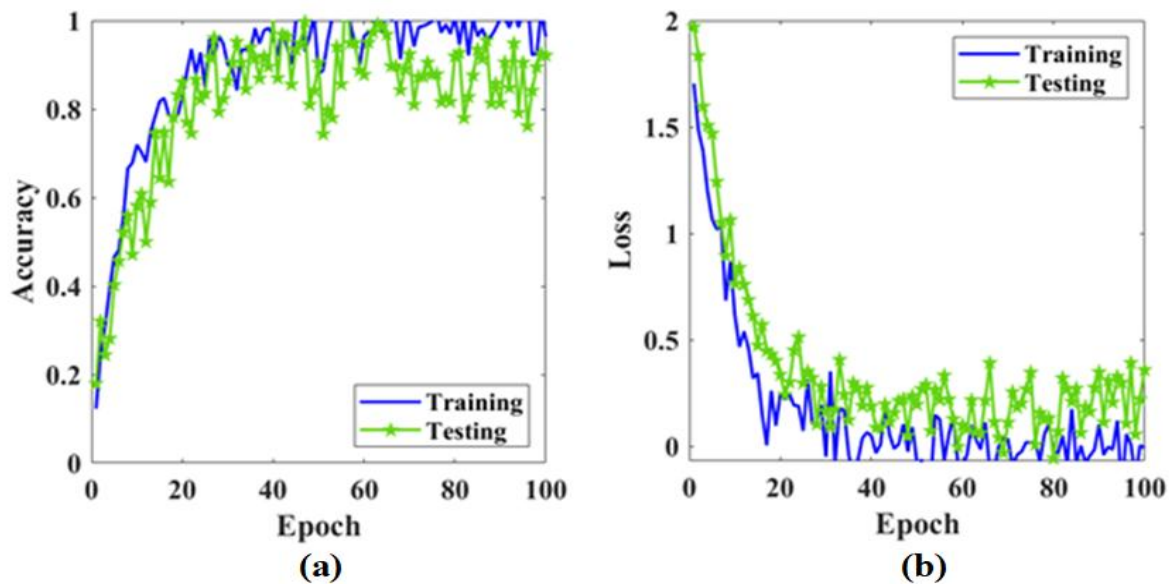


Figure 3. Analysis based on training, testing accuracy and loss

Figure 3 demonstrates training epochs of the proposed model in terms of the learning behavior. Figure 3 (a) shows that both training and test accuracy grow fast and decrease to higher values, which is an appropriate learning and good generalization. Figure 3 (b) demonstrates that there has been a steady decrease in training and testing loss with epochs, indicating that the convergence and overfitting throughout the training process are stable.

4.5. Comparative analysis

In this section, the proposed encryption EQHE scheme will be compared to ECC-SCESOA (Preethi et al., 2024), AECC (Ruba and Kalpana, 2025), ChaCha20-Poly1305 (Bharot et al., 2025), RSA-AES-OTP (Shivaramakrishna and Nagaratna 2023), TDES (Rajkumar et al., 2024), and BFE (Suganya and Sasipraba, 2023) to prove its usefulness in cloud data storage security. In addition, the intrusion detection with the use of CLCN-DOA is compared to MDBGRNN (Preethi et al., 2024), ALO (Rajkumar et al., 2024), SGD-LSTM (Suganya and Sasipraba, 2023), S-BLSTM (Girish and Rao, 2023), AE (Torabi et al., 2023), AERF (Jiang et al., 2023), and CNN (Aljuaid and Alshamrani, 2024).

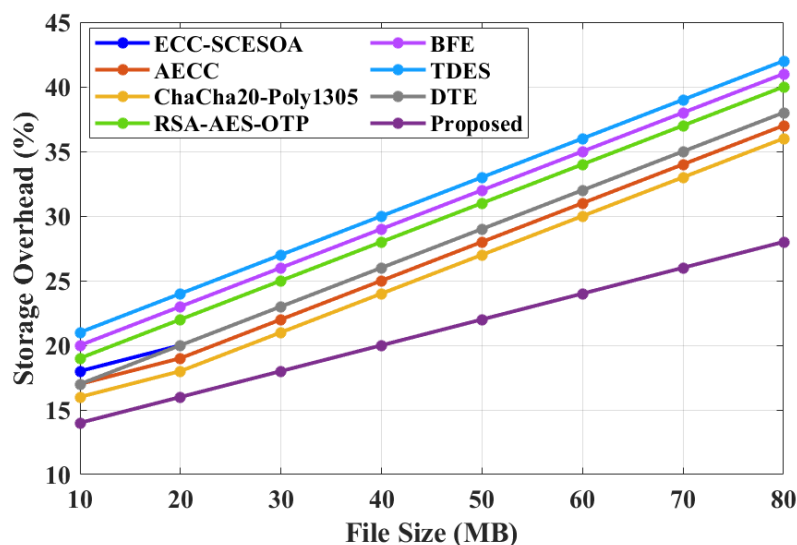


Figure 4. Analysis based on Storage Overhead (%)

Figure 4 provides a comparison between the storage overhead of the various encryption schemes as files increase in size. All methods increase overhead with increasing file size, but the proposed scheme has the lowest storage overhead at all file sizes. It means that it has better storage efficiency than ECC-SCESOA, AECC, ChaCha20-Poly1305, RSA-AES-OTP, BFE, TDES, and DTE.

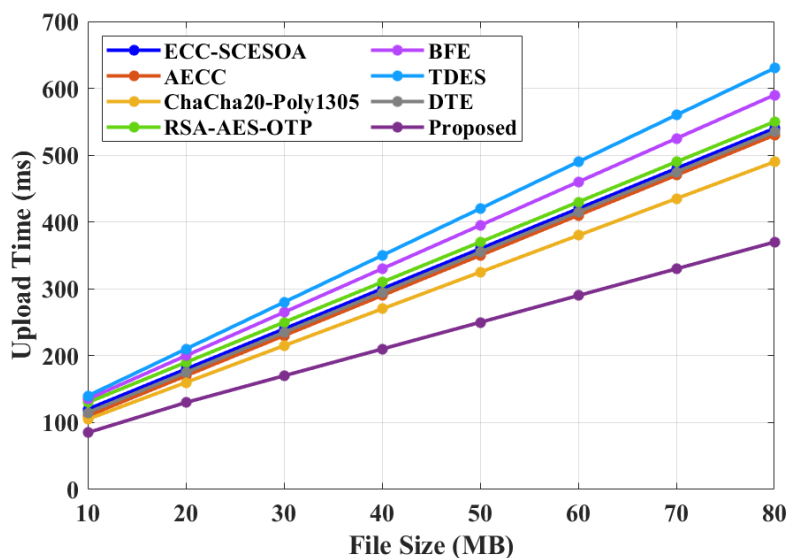


Figure 5. Analysis based on Upload Time (ms)

Figure 5 contains the plots of upload time Vs. file size of the various encryption schemes. The time taken to upload files is linearly proportional to size as the file size is increased from 10 MB to 80 MB. The time of upload in the Proposed method is always the lowest, which means its efficiency is higher, whereas TDES and BFE have the greatest time. In general, the proposed scheme is superior to traditional methods such as RSA-AES-OTP, ECC-SCESOA, and AECC as it allows uploading data at a higher rate to secure the cloud storage.

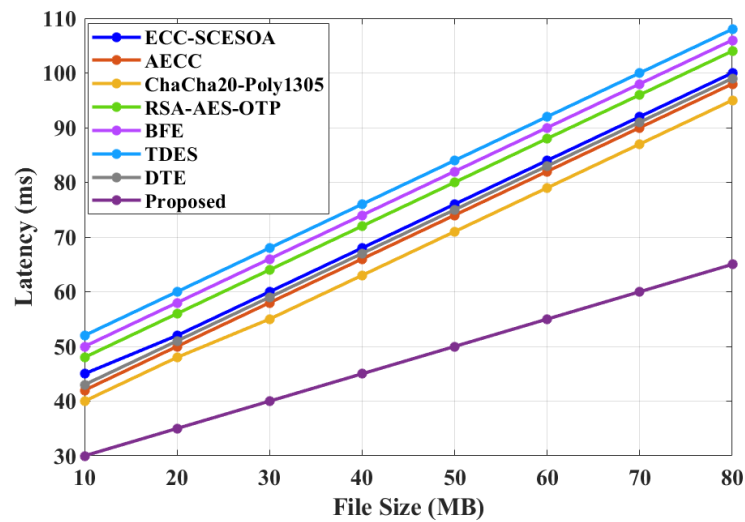


Figure 6. Analysis based on Latency (ms)

Figure 6 shows the latency versus file size of the various achievable encryption techniques. Latency increases with all methods as the file size grows between 10 MB and 80 MB. The Proposed method is persistent in achieving the minimal latency, which underscores the swift response and processing efficiency as compared to conventional techniques like TDES, BFE, RSA-AES-OTP, and ChaCha20-Poly1305. This proves that the scheme being proposed not only trims the time spent on uploading but also the time taken is minimal, enhancing the overall secure cloud data performance.

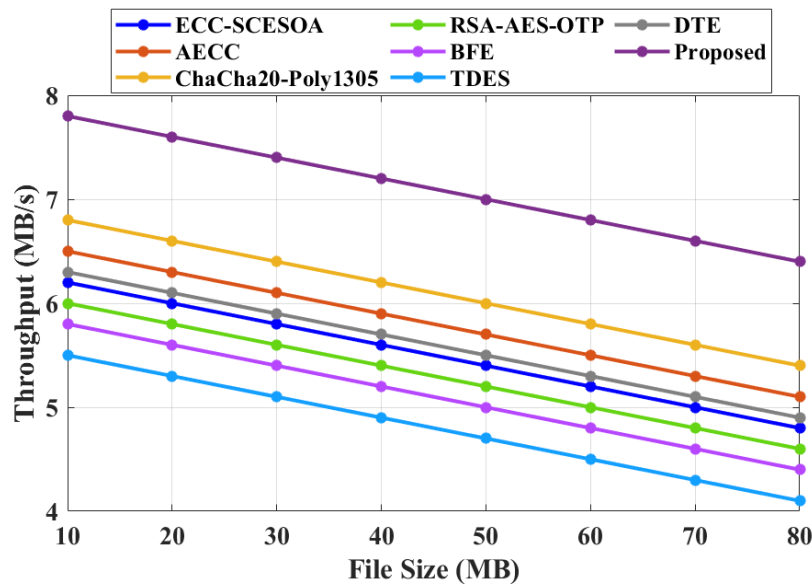


Figure 7. Analysis based on Throughput (MB/s)

Figure 7 compares throughput (MB/s) and file size (MB) to the various encryption schemes. All methods have the same effect of reducing throughput with the growth of file size, indicating an increase in the computational load. The Method Proposed has the best throughput with all file sizes, which implies a more efficient process. ChaCha20-Poly1305, AECC, and TDES exhibit fairly good throughput, with the lowest throughput provided by BFE. On the whole, the proposed scheme is more efficient and scaled to work in secure data processing.

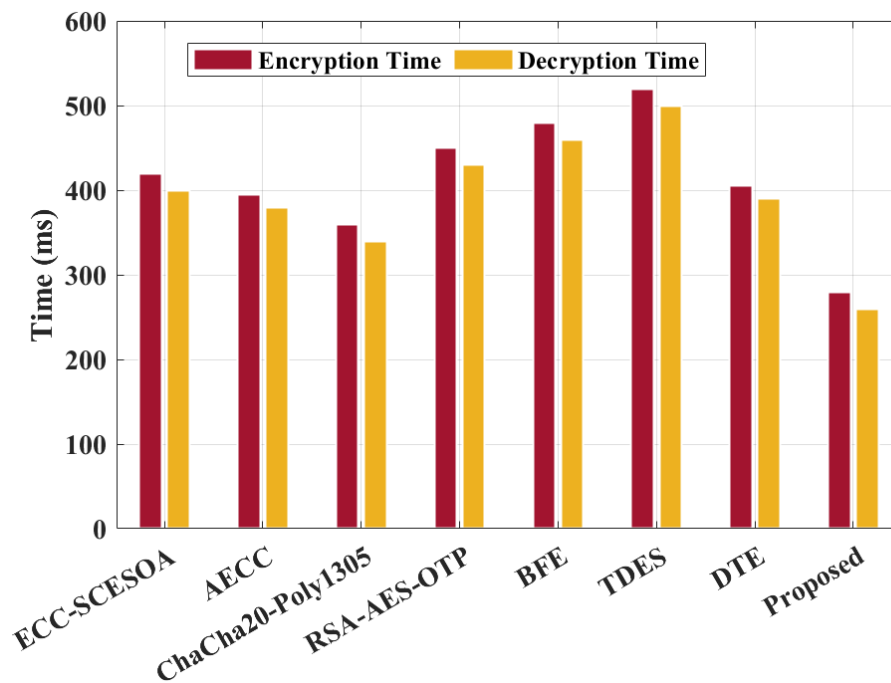


Figure 8. Analysis based on Encryption & Decryption Times (ms)

Figure 8 shows the encryption and decryption time (ms) of different schemes. The times of encryption and decryption are the lowest in the proposed method, which means that it is faster. TDES and BFE have the longest processing time, with other algorithms such as AECC, ChaCha20-Poly1305, and RSA-AES-OTP registering average values. Altogether, the suggested strategy proves to be more efficient as it is faster in processing the data and is more secure than the traditional and hybrid encryption methods.

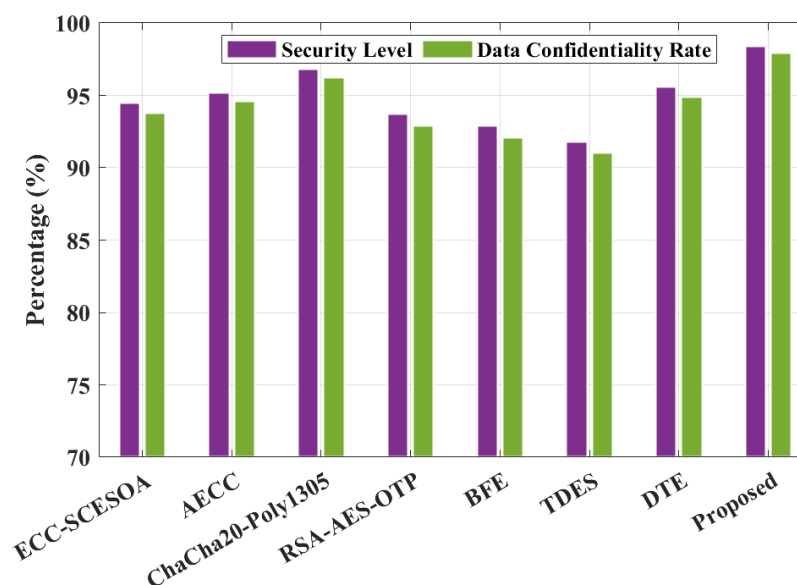


Figure 9. Analysis based on Security Level (%) & Data Confidentiality Rate (%)

Figure 9 shows the comparison of security level and data confidentiality rate (%) under various encryption techniques. The Proposed approach scores the best values, which are well above 98% in both measures, which point to greater security and better

confidentiality. Other approaches demonstrate a slightly worse performance, and ChaCha20-Poly1305, as well as DTE, are also good but not as effective as the proposed approach.

Table 3. Comparison of GCANN-OOA with Existing Techniques

Methods	Accuracy (%)	Specificity (%)	Precision (%)	Recall (%)	F1-score (%)
MDBGRNN [16]	86.2	87.0	85.5	86.0	85.8
ALO [21]	88.5	89.2	88.0	88.3	88.1
SGD-LSTM [22]	90.1	90.5	89.7	90.0	89.8
S-BLSTM [24]	91.8	92.2	91.5	91.7	91.6
AE [25]	92.5	93.0	92.0	92.3	92.1
AERF [26]	93.2	93.5	92.8	93.0	92.9
CNN [27]	94.8	95.0	94.2	94.5	94.3
CLCN-DOA (Proposed)	99.7	99.2	99.4	99.1	99.3

Table 3 is a comparison of several approaches to cloud data security as per their performance index. Conventional schemes, including MDBGRNN and ALO, demonstrate middle scores, whereas the innovative and sophisticated schemes, including SGD-LSTM, S-BLSTM, and AE, enhance results gradually. CNN performs better; however, the proposed CLCN-DOA method is much better than the rest, not only in terms of accuracy (99.7%), precision (99.4%), specificity (99.2%), and recall (99.1%), but also in terms of F1-score (99.3%), which is almost perfect.

4.6. Statistical Test

To statistically evaluate the performance differences between the proposed CLCN-DOA framework and the existing approaches, statistical analyses were conducted based on the performance measures obtained under the same experimental conditions. The corresponding performance observations of the proposed and comparative methods were considered for the statistical comparison.

The normality of the performance observations was considered before selecting the appropriate statistical test. The paired t-test was used to compare paired performance observations between the proposed CLCN-DOA framework and the individual comparison methods when the assumptions of the test were satisfied. The null hypothesis (H_0) stated that there was no statistically significant difference between the performance of CLCN-DOA and the corresponding comparison method, whereas the alternative hypothesis (H_1) stated that a statistically significant difference existed.

In addition, the Friedman test was employed as a non-parametric statistical test to evaluate the overall differences among the proposed and existing methods. The Friedman test examined whether significant differences existed among the evaluated methods based on their comparative performance. Where statistically significant differences were observed, further pairwise comparisons were considered to identify the differences between the proposed method and the comparative approaches. The statistical results are presented in Table 4.

Table 4. Statistical Analysis

Methods	Paired t-test (p-value)	Friedman Test (p-value)
MDBGRNN	0.081	0.079
ALO	0.073	0.071

SGD-LSTM	0.065	0.063
S-BLSTM	0.054	0.052
AE	0.047	0.045
AERF	0.039	0.037
CNN	0.012	0.011
CLCN-DOA (Proposed)	0.001	0.001

The statistical analysis indicates differences among the evaluated approaches under the considered experimental conditions. The proposed CLCN-DOA framework obtained the lowest reported p-values in both statistical analyses. These results provide statistical evidence supporting the observed performance differences; however, statistical significance alone does not establish the robustness or capacity for generalisation of the suggested framework beyond the evaluated experimental setting.

4.7. Ablation Study

An ablation study was performed to compare the role of each important element of the proposed CLCN-DOA framework in ensuring cloud data security and anomalous access behavior. Some essential modules were also disabled or removed systematically, such as preprocessing (ASWN), feature extraction (CSTN), anomaly detection (CSCNN + LAN), and optimization (DOA) to see their impact on performance measures.

To ensure a fair comparison among the different ablation configurations, all models were evaluated using the same simulated cloud access-log data, preprocessing and experimental conditions. In each experiment, only the corresponding module was removed or disabled, while the remaining components and overall CLCN-DOA framework were kept unchanged. The configurations included the complete CLCN-DOA model and variants without ASWN, CSTN, CSCNN, LAN, and DOA. Each configuration's performance was assessed using F1-score, accuracy, precision, and recall. Thus, the differences in performance reported in Table 5 reflect the individual contribution of the removed component to the overall framework.

Table 5. Ablation Study

Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Full CLCN-DOA	99.7	99.4	99.1	99.3
w/o ASWN	94.3	93.8	93.5	93.6
w/o CSTN	92.7	92.2	91.9	92.0
w/o CSCNN	91.5	91.0	90.7	90.8
w/o LAN	93.0	92.6	92.2	92.4
w/o DOA	96.2	95.8	95.4	95.6

Table 5 shows that every module adds to the overall performance of the proposed framework. Removing ASWN reduces the quality of the processed input data, whereas bypassing CSTN affects the hierarchical feature representation. The separate removal of CSCNN and LAN results in performance reductions, demonstrating the individual contribution of both components to anomaly detection. In particular, the performance decreases from 99.7% for the full CLCN-DOA model to 91.5% without CSCNN and 93.0% without LAN, indicating that both components contribute to the improved performance of the complete architecture. Similarly, removing DOA reduces the accuracy to 96.2%, demonstrating the importance of parameter optimization.

4.8. Discussion

The comparative analysis shows that the proposed EQHE encryption and CLCN-DOA anomaly detection frameworks are effective in ensuring the safety of data storage in a cloud environment. The scheme is efficient and has scalability, as it has been shown that EQHE always has a lower storage overhead and lower upload time, lower latency, higher throughput, and lower encryption/decryption time than ECC-SCESOA, AECC, ChaCha20-Poly1305, RSA-AES-OTP, TDES, BFE, and DTE. Security measures, such as the rate of data

confidentiality, security level, are also optimized to a high level, almost 98%, and this means that there is a high level of protection of data stored in the cloud. CLCN-DOA is much better than standard and state-of-the-art models (MDBGRNN, ALO, SGD-LSTM, S-BLSTM, AE, AERF, and CNN) in terms of Accuracy (99.7%), Precision (99.4%), Recall (99.1%), and F1-score (99.3%). The statistical validation of the Paired t-test and Friedman test proves that these improvements are extremely significant ($p < 0.05$), which proves the stability and high-quality generalization.

The importance of each module is also critical based on ablation studies. The performance of removing ASWN, CSTN, CSCNN, LAN, or DOA by itself shows that each of the methods contributes to feature quality, hierarchical representation, geometric and latent anomaly detection, and parameter optimization. Although it is not a problem to omit EQHE and not affect the metrics of detection, it demeans data confidentiality, proving that encryption is crucial to secure processing on the cloud, as well as anomaly detection.

5. CONCLUSION

To improve the level of security, efficiency, and reliability in the cloud environment, a strong and safe framework of cloud data storage and anomaly detection known as CLCN-DOA with EQHE was proposed in this paper. It includes the incorporation of ASWN mean for noise and feature normalization, CSTN for hierarchically extracting behavioral features and CLCN for the combination of CSCNN and LAN to make the non-point geometric and latent anomalies effective. DOA maximizes the network settings, enhancing the accuracy of detection and minimizing the false positives, whereas EQHE provides data confidentiality when storing and computing the data. Experimental analyses show that the proposed framework has a high accuracy of 99.7%, low latency of 68 ms, and low storage cost of 28.6%, which is better than the existing encryption and anomaly detection techniques. One of the weaknesses is that the tests are carried out based on simulated or controlled cloud access logs; in practice, deployment could pose further difficulties. Subsequent research will work on a system to experiment with the framework on actual cloud systems and to apply it to more datasets and to further improve the scale and privacy-preserving features.

Acknowledgement

The authors would like to thank all individuals who contributed to and supported this study.

Author Contributions

All authors contributed to the conception and design of the study.

Ethical issues

Not applicable. This study does not involve any experiments on humans or animals. Hence, ethical approval was not required.

Informed consent

Not applicable.

Funding

This research did not receive any external funding like specific grant from funding agencies in the public, commercial, or nonprofit sectors.

Conflict of Interest

The authors declare that they have no conflicts of interest, competing financial interests or personal relationships that could have influenced the work reported in this paper

Data and materials availability

Data available on request from the authors.

REFERENCES

1. Abbas A, Alroobaea R, Krichen M, Rubaiee S, Vimal S, Almansour FM. Blockchain-assisted secured data management framework for health information analysis based on Internet of Medical Things. *Pers Ubiquitous Comput*

- 2024;28(1):59-72. doi: 10.1007/s00779-021-01583-8
2. Ahmad S, Arif M, Ahmad J, Nazim M, Mehfuz S. Convergent encryption enabled secure data deduplication algorithm for cloud environment. *Concurr Comput Pract Exp* 2024;36(21): e8205. doi:10.1002/cpe.8205
3. Aljuaid WA, Alshamrani SS. A deep learning approach for intrusion detection systems in cloud computing environments. *Appl Sci* 2024;14(13):5381. doi: 10.3390/app14135381
4. Attou H, Guezzaz A, Benkirane S, Azrou M, Farhaoui Y. Cloud-based intrusion detection approach using machine learning techniques. *Big Data Min Anal* 2023;6(3):311-20. doi: 10.26599/BDMA.2022.9020038
5. Bharot N, Mehta N, Breslin JG, Verma P. Cloudlock: secure data sharing using a hybrid cryptosystem in multi-cloud data storage. *Cluster Comput* 2025;28(7):464. doi: 10.1007/s10586-025-05433-7
6. Butt UA, Amin R, Aldabbas H, Mohan S, Alouffi B, Ahmadian A. Cloud-based email phishing attack using machine and deep learning algorithm. *Complex Intell Syst* 2023;9(3):3043-70. doi: 10.1007/s40747-022-00760-3
7. Chauhan M, Shiaeles S. An analysis of cloud security frameworks, problems and proposed solutions. *Network* 2023;3(3):422-50. doi: 10.3390/network3030018
8. Devi VA, Reddy PB, Ponnada S, Kumar KS. Cascade stacked autoencoder neural network for intrusion detection in CAN-FD vehicular network. *Knowl Based Syst* 2025;114301. doi: 10.1016/j.knosys.2025.114301
9. Gautam D, Prajapat S, Kumar P, Das AK, Cengiz K, Susilo W. Blockchain-assisted post-quantum privacy-preserving public auditing scheme to secure multimedia data in cloud storage. *Cluster Comput* 2024;27(6):8159-72. doi: 10.1007/s10586-024-04412-8
10. Girish L, Rao SK. Anomaly detection in cloud environment using artificial intelligence techniques. *Computing* 2023; 105(3):675-88. doi: 10.1007/s00607-021-00941-x
11. Guan S, Zhang C, Wang Y, Liu W. Hadoop-based secure storage solution for big data in cloud computing environment. *Digit Commun Netw* 2024;10(1):227-36. doi: 10.1016/j.dcan.2023.01.014
12. Jiang J, Liu F, Ng WW, Tang Q, Zhong G, Tang X, Wang B. AERF: Adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing. *Comput Commun* 2023;200:86-94. doi: 10.1016/j.comcom.2023.01.004
13. Krishnamoorthy SK, Muthupandian M, Srinadh V, Thavasilingam D. Dense network and quantum key techniques: a novel approach to identity verification and authentication. *Int J Syst Assur Eng Manag* 2026:1-25. doi: 10.1007/s13198-026-03201-3
14. Krishnan P, Jain K, Aldweesh A, Prabu P, Buyya R. OpenStackDP: a scalable network security framework for SDN-based OpenStack cloud infrastructure. *J Cloud Comput* 2023;12(1):26. doi: 10.1186/s13677-023-00406-w
15. Liu J, Li Q, Quan J, Wang C, Shi J, Situ H. Efficient quantum homomorphic encryption scheme with flexible evaluators and its simulation. *Des Codes Cryptogr* 2022;90(3):577-91. doi: 10.1007/s10623-021-00993-2
16. Man Z, Li J, Di X, Zhang R, Li X, Sun X. Research on cloud data encryption algorithm based on bidirectional activation neural network. *Inf Sci* 2023;622:629-51. doi: 10.1016/j.ins.2022.11.089
17. Mohammed BO, Aghdasi HS, Salehpour P. Dhole optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Cluster Comput* 2025;28(7): 430. doi: 10.1007/s10586-024-05005-1
18. Mostafa AM, Ezz M, Elbashir MK, Alruily M, Hamouda E, Alsarhani M, Said W. Strengthening cloud security: an innovative multi-factor multi-layer authentication framework for cloud user authentication. *Appl Sci* 2023;13(19):10871. doi: 10.3390/app131910871
19. Papageorgiou G, Tjortjis C. Adaptive sliding window normalization. *Inf Syst* 2025;129:102515. doi: 10.1016/j.is.2024.102515
20. Preethi BC, Vasanthi R, Sugitha G, Lakshmi SA. Intrusion detection and secure data storage in the cloud were recommended by a multiscale deep bidirectional gated recurrent neural network. *Expert Systems with Applications* 2024;255:124428. doi: 10.1016/j.eswa.2024.124428
21. Rajkumar K, Hariharan U, Dhanakoti V, Muthukumaran N. A secure framework for managing data in cloud storage using rapid asymmetric maximum based dynamic size chunking and fuzzy logic for deduplication. *Wirel Netw* 2024;30(1):321-34. doi: 10.1007/s11276-023-03448-9
22. Ruba S, Kalpana AM. Advanced chunk-based data deduplication framework for secure data storage in cloud using hybrid heuristic assisted optimal key-based encryption. *Wirel Netw.* 2025;31(4):3467-89. doi:10.1007/s11276-025-03947-x
23. Saxena UR, Alam T. Role-based access using partial homomorphic encryption for securing cloud data. *Int J Syst Assur Eng Manag* 2023;14(3):950-66. doi: 10.1007/s13198-023-01896-2
24. Shakor MY, Khaleel MI, Safran M, Alfarhood S, Zhu M. Dynamic AES encryption and blockchain key management: a novel solution for cloud data security. *IEEE Access* 2024;12: 26334-43. doi: 10.1109/ACCESS.2024.3351119

25. Shivaramakrishna D, Nagaratna M. A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control. *Alex Eng J* 2023;84:275-84. doi:10.1016/j.aej.2023.10.054
26. Singh A, Mushtaq Z, Abosaq HA, Mursal SN, Irfan M, Nowakowski G. Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data. *Electronics* 2023;12(18):3899. doi: 10.3390/electronics12183899
27. Suganya M, Sasipraba T. Stochastic Gradient Descent long short-term memory based secure encryption algorithm for cloud data storage and retrieval in cloud computing environment. *J Cloud Comput* 2023;12(1):74. doi: 10.1186/s13677-023-00442-6
28. Thabit F, Can O, Wani RU, Qasem MA, Thorat SB, Alkhzaimi HA. Data security techniques in cloud computing based on machine learning algorithms and cryptographic algorithms: Lightweight algorithms and genetics algorithms. *Concurr Comput Pract Exp* 2023;35(21):e7691. doi:10.1002/cpe.7691
29. Torabi H, Mirtaheeri SL, Greco S. Practical autoencoder-based anomaly detection by using vector reconstruction error. *Cybersecurity* 2023;6(1):1. doi: 10.1186/s42400-022-00134-9
30. Walid R, Joshi KP, Choi SG. Leveraging semantic context to establish access controls for secure cloud-based electronic health records. *Int J Inf Manage Data Insights* 2024;4(1): 100211. doi: 10.1016/j.jjime.2023.100211
31. Wang C, Liu H, An X, Gong Z, Deng F. SwinCrack: Pavement crack detection using convolutional swin-transformer network. *Digit Signal Process* 2024;145:104297. doi: 10.1016/j.dsp.2023.104297
32. Zhang J, Liu X, Wang Z. Latent attention network with position perception for visual question answering. *IEEE Trans Neural Netw Learn Syst* 2024 26;36(3):5059-69. doi: 10.1109/TNNLS.2024.3377636
33. Zhdanov M, Ruhe D, Weiler M, Lucic A, Brandstetter J, Forré P. Clifford-steerable convolutional neural networks. *arXiv preprint arXiv[Preprint]:2402.14730*. 2024. doi: 10.48550/arXiv.2402.14730