Journal of Information Systems Engineering and Management
2025, 10(12s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Secure and Efficient Cloud Data Retrieval Using Privacy-Preserved Hybrid CRNN with Swallow Swarm Optimization

D Kalpana[1*], Dr. K Ram Mohan Rao[2]
[1] *Jyothishmathi Institute of Technology and Science, Telangana, India*
[2] *Vasavi College of Engineering. Hyderabad, Telangana, India*
E-mail: kalpanadev@gmail.com, krmrao@staff.ac.in

ARTICLE INFO

Received: 20 Nov 2024
Revised: 31 Dec 2024
Accepted: 19 Jan 2025

ABSTRACT

The application of Cloud Computing (CC) has increased popularity in recent years. This technology allows for resource sharing and extensive capabilities, making it feasible to store and analyze data remotely on the cloud. However, it is not secured, some parties can access to a network like the internet and read or alter data, making this cloud untrustworthy. Consequently, one of the issues that must be resolved while utilizing CC involves preserving data security and privacy. Several strategies based on various Encryption (Enc) systems have been explored to address data privacy and integrity. Cloud-related risks include data loss and leakage, malware attacks, and exploited vulnerabilities. In order to prevent attacks and to preserve privacy when keeping data in the cloud, it is crucial to make sure that a foolproof protecting system is in operation. In order to entirely understand the value of Medical Data (MD) and realize data collaborative sharing, a Deep Learning (DL) architecture that uses Homomorphic Encryption (HE) technology was established in this work to protect training parameters and created a MD security sharing scheme based on the communication mode.

In this case, the training parameters are protected by using the Paillier HE (PHE) to achieve additive homomorphism. This article offers a Privacy-Preserving (PP) hybrid Convolutional Recurrent Neural Network (CRNN) based on Swallow swarm optimization technique (SSO) to address the issue of privacy leaking. By combining DL with HE, a knowledge transfer strategy with PP is created. Researchers may infer from the simulation results that Learning Rate (LR), batch size, and other factors are connected to the model prediction accuracy. The outcomes demonstrate that this method has good performance, completes the accurate disease prediction, and accomplishes Data Sharing (DS) while maintaining data privacy.

Keywords: Cloud Computing (CC), Paillier Homomorphic Encryption (PHE), Privacy Preservation (PP), Enhanced Recurrent Neural Network (ERNN) based Swallow Swarm Optimization algorithm (SSO).

1. Introduction

In current era of modern and digital communications, cloud models and cloud services (CS) are becoming extremely crucial. Furthermore, infrastructure and software issues are provided by the cloud model. A few cloud services that employ user location, identity, and private data are Cloud Sae, Amazon's basic storage service, and others [1]. The most significant issue in recent cloud services has been the way cloud models handle various data security and privacy issues. Customers and cloud users are the ones whose primary entitlement to data security is to save their private information, such as bank account details, health information, and so forth. Secure protocols, signature schema, and unknown user authentication methods are a few examples of cryptography devices and models.

Critical information is protected against unauthorized access using User Authentication (UA). For instance, User B's sensitive information is not accessible to User A, who only has relevant information. If UA is not secure, hackers can hack a system and access data [2]. A secure protocol that permits two networked peers to communicate safely is provided by Secure Sockets Layer (SSL) and Transport Layer Security (TLS). When sending private data from a Web Browser (WB) to a web server (WS), SSL is most frequently utilized. Sensitive and confidential data, such

credit card numbers, passwords, and private messages, benefit greatly from TLS encryption, which prevents hackers and unauthorized individuals from reading data transferred over the Internet.

In order to grant users authorized access, the CSPs must be managed through the authentication procedure [3]. It is the basis for revoking the identities of the malicious users. Users use cloud resources with effective dynamic reliability and service integration procedure according to their needs. User privacy must be protected, processed using a secure paradigm, and made available to authorized healthcare providers in the context of medical data processing. There is still significant expense and computational complexity even with the various Enc and key distribution mechanisms that are created in the current secure cloud Data Sharing (DS) technologies.

For converting Plain Text (PT) data (PT) into something that appears random and unimportant (Cipher Text (CT)) and this procedure is called Enc . The process of converting CT to PT is called Dec [4]. Many pieces of information have been combined by the use of Symmetric Encryption (SE). The virtualization of currently accessible data services or data centers, which offers a multipurpose framework and supports services to various clients, is referred to as CC. Many services available over the Internet, including servers, databases, networking, software, and data storage, are sourced from CC. Cloud storage makes it possible to store files to a remote database and retrieve them whenever needed. Reliability, pay-per-use pricing, and accessibility are the primary advantages of CC [5].

Moreover, CC is considered to have drawbacks in terms of security, recurring expenses, and decreased control over infrastructure and flexibility. Services in the cloud are rendered according to the needs of the user. The cloud model also offers effective service to users with a dependable CS model, and it saves large amounts of data. These days, users connect to and use cloud-based programs over the Internet with Software (aaS) as a service (SaaS), along with the current services like Infrastructure aaS (IaaS), Platform aaS (PaaS). New cloud models like Supercomputing aaS (SCaaS) and High-Performance Computing aaS (HPCaaS) are also made possible by the effectiveness of new technologies.

Microsoft Office 365 provides additional popular instances. such as calendaring, email, and office tools. Paying for the entire software system solution on a "pay-as-you-go (PAYG)" basis is possible with Clouds Service Providers (CSP). The term "PAYG" CC service that provides on-demand networking, storage, and processing power is called IaaS. SaaS, PaaS, serverless, and IaaS are the four categories of cloud services [7]. From basic cloud-based applications to sophisticated Cloud-Based (CB) organizations, PaaS is an all-inclusive cloud development and deployment environment.

Encrypting the data before outsourcing is a simple way for users to protect the privacy of their data. The Amazon IS service has adapted this model. Keyword-based searches on encrypted data become challenging even while privacy is maintained [8]. To search over PT, a naive method needs the data to be downloaded and decrypted. Such an approach severely reduces the benefits of using the cloud by producing enormous overheads in processing and communication. Though the computing overhead is so high that it is not at all practical, completely HE permits arbitrary operations, such as searches over encrypted data [9].

Due to the scalable and affordable services provided by CSP, CC has been seen as an appropriate platform to deploy standard MD systems by healthcare providers who are looking to automate processes of health information manipulation at lower costs and higher gains. Though CB DS systems are becoming more popular, their adoption in the medical sector has been hindered by privacy-related issues. Over the past few years, a great deal of research has been done on the many security and privacy issues pertaining to clinical information [10]. Reports on the security and privacy concerns related to the manipulation of clinical information in networked systems have been released by numerous organizations.

The European Data Protection Directive 95/46/EC and the Health Insurance Portability and Accountability Act (HIPAA) are the two most often used regulations. PP during transmission and PP of the stored data are the two main concerns about the privacy of Medical DS (MDS) that these regulations address. The SSL and TSL protocols both address the former, which has been the subject of extensive study. The latter is more pertinent to storage as a service under the CC framework, because the data that is outsourced is kept on the CSP's website, but it has received less research [11]. Several strategies based on various Enc systems have been explored to address data privacy and integrity.

Cloud-related risks include malware attacks, data loss and leakage, and exploited vulnerabilities. Thus, it is essential to make sure that anonymity is preserved when keeping data in the cloud and that a foolproof security system is in place to prevent attacks. In order to fully realize the value of medical data and actualize data collaboration sharing, this research effort constructed a DL framework that uses HE technology to safeguard training parameters and created a MD security sharing scheme based on the communication mode.

Following shows the arrangement of the remaining study: A few of the modern techniques for a PP CC security architecture are examined in Section 2. The suggested methodology's approach is presented in section 3. The results and the discussion are given in section 4. Future work and the conclusion are covered in section 5.

2. Literature Review

In this section, an in-depth analysis of the associated solutions for security concerns, cloud data services, and multiple risks are provided.

A service-oriented query (SOQ) approach was presented by Song et al. [12]; it adaptively modifies the encrypted data buckets according to the workload of queries and the distribution of sensitive data. In order to get around the still-unresolved join query issue between Enc attributes, also suggest a two-stage indexThe strategies achieve effective encrypted data query performances, as demonstrated by the experiments designed to assess the performance of the suggested method .

To verify data security and integrity for data outsourcing in cloud environments, the PP Data Security Approach (PP-DSA) was suggested by Kirubakaran et al. [13]. The Efficient Authentication Technique (EAT) in conjunction with the Third-Party Auditor (TPA) Group Signature approach ensures PP in this work. Data security and shared data integrity are the auditor's two main responsibilities. Furthermore, the attackers that need to be dealt with by the EAT may also be the CSP and Data User (DU). Improving cloud security and enhancing Quality of Service (QoS) is the primary objective of this research. The suggested model outperforms previous approaches when compared to the results, which are assessed based on the model's efficacy, security, and dependability.

A practical strategy for CC MDS that protects privacy was put out by Yang et al. [14]. To account for various medical datasets with varying privacy issues, employing a vertical partition of medical dataset based on a classification of clinical record features. In order to provide multiple paradigms of balance between medical data utilization and PP, it primarily consists of 4 components:

(1) MD publishing vertical data partition
(2) Data fusion for accessing medical datasets
(3) Verification of integrity

and(4) hybrid search utilizing both CT and PT

For the large-scale access and sharing of medical data, a prototype system is put into operation.

A unique Blockchain (BC)-assisted framework for efficient cloud platform DS and Data retrieval was developed by Gajmal et al. [15]. For secure transmission, the EHR application has developed a data protection model. The Inter-Planetary File System (IPFS), smart contracts, transactional BC, DU, and data owner are among the entities on the cloud platform. By transferring the secured EHR to IPFS before sharing it with the DU, the data owner in this case incorporates a data protection framework to secure EHR. The suggested Conditional Autoregressive Value at risk (CAViaR)-based (BSO) Bird swarm algorithm for producing optimal PP coefficients is used to secure data privacy. Considering utility and privacy, a new Objective Function (OF) was created. With least responsiveness of 251.339 s, maximal real user detection of 32.451%, maximal privacy of 96.5%, and minimal information loss of 3.5%, the suggested CAViaR-based BSA outperforms other techniques.

A parallel retrieval approach that is adaptable to such a structure has been suggested by Wang et al. [16] along with a parallel Binary Search Tree (BST) structure constructed in block format. The approach performs better in searches, according to a quantitative investigation using the information retention index. Furthermore, the unexplainability of the feature vectors produced by the approach makes reverse analysis challenging, improving patient and researcher PP. The EEPR technique delivers a lower time complexity and greatly increases both search

efficiency and accuracy over existing schemes, as demonstrated by a formal security research. It is also resistant to known background attacks.

For the outsourced searchable encrypted data, A PP STorage and REtrieval (STRE) technique has been suggested by Li et al. [17]; it guarantees dependability and security while also maintaining privacy. Users of cloud services can distribute and search their encrypted data throughout several separate clouds managed by various CSPs due to the STRE mechanism, which remains stable even in the event that a specific number of CSPs fail. STRE has the advantage of a partially hidden search pattern in addition to reliability. The approach efficacy and efficiency are demonstrated by the real-world dataset by assessment of the STRE mechanism on Amazon EC2.

A high-level review of the related security technologies is provided after Tang et al. [18] developed a security architecture for outsourcing data services to the cloud. Then, cloud data services such as data search, data processing, DS, data storage, and data access are offered by a focus on current security measures, that are safe, dependable, and private. At last, for each category of solutions, suggesting an open challenges and possible research areas.

Order-preserving SE (OPSE), an established cryptographic technique, is effectively utilized in Wang et al.'s formulation of ranked searchable SE [19]. A detailed examination reveals that the suggested approach accurately achieves the aim of ranked keyword search while having a "as-strong-as-possible" security assurance in comparison to earlier SSE systems. The effectiveness of the suggested solution is demonstrated by extensive research results.

Basic notion for the MRSE based on secure inner product computation was suggested by Cao et al. [20] to meet a variety of strong privacy criteria in two distinct threat models. Two greatly enhanced MRSE approaches were subsequently presented. Extend these two techniques further to allow additional search semantics in order to enhance the data search service's search experience. A detailed examination of the efficiency and privacy guarantees of the suggested methods is provided. Real-world dataset experiments confirm that the suggested approaches offer minimal computational and communication overhead.

A unique PP functional encryption-based search mechanism for encrypted cloud data was created by Liang et al. [21]. When compared to the current public key based search systems, one of the main advantages of the new primitive is its support for regular language search, which is an extremely expressive search mode. The suggested system is demonstrably safer and more effective than several searchable systems with high expressiveness, according to the security and performance studies.

A "Greedy (DFS) Depth-first Search" technique was introduced by Xia et al. [22] to offer effective multi-keyword ranked search. The Index Vectors (IV) and Query Vectors (QV) are encrypted using the secure kNN method, which also guarantees precise relevance score computation between the encrypted IV and QV. To protect against statistical attacks, phantom phrases are inserted to the IV to blind search results. The suggested technique can achieve sub-linear search time and handle document deletion and insertion flexibly since it makes use of the unique tree-based index structure. In-depth tests are carried out to show that the suggested plan is effective.

3. Proposed Methodology

This study developed an MD security sharing method based on the communication mode and a DL framework that uses HE technology to secure training parameters. In this case, the training parameters are protected by using the PHE to achieve additive homomorphism. This article offers a PP hybrid CRNN based on SSO to address the issue of privacy leaking. By merging DL with HE, a knowledge transfer strategy with PP is created.

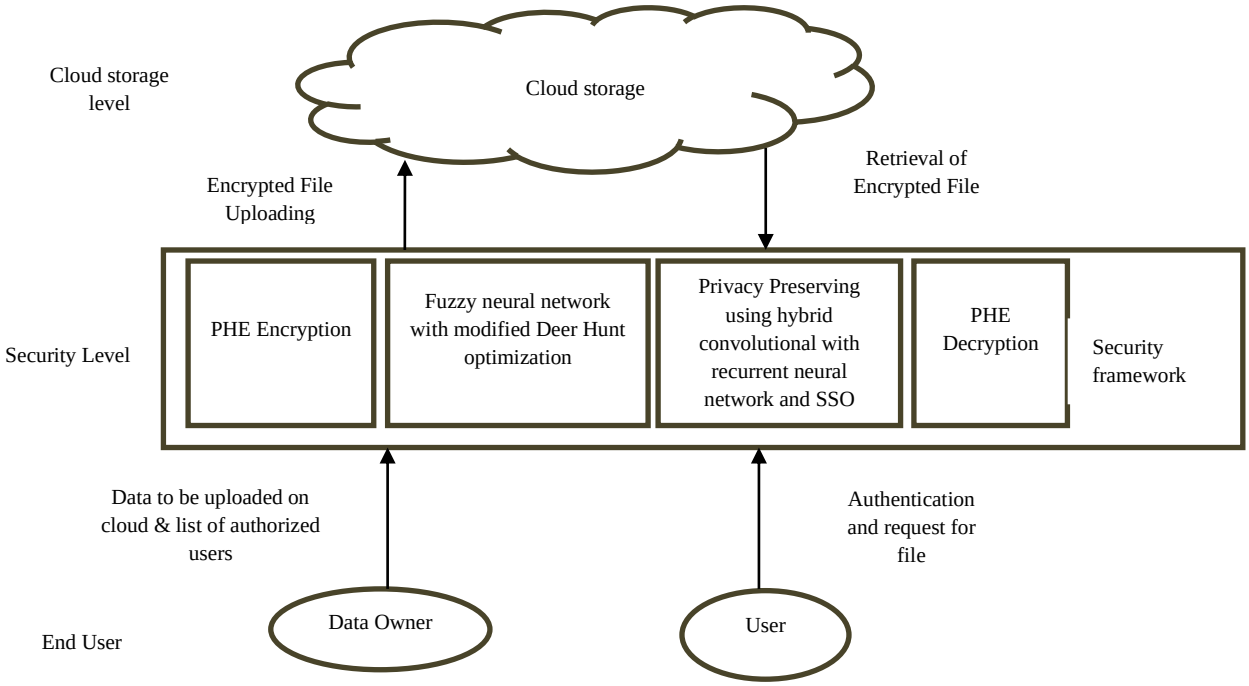

figure 1. The procedure of the suggested method

3.1. Security Model

The entire system security structure model has been separated into three layers, which are depicted in Figure 1.

(a) Level of the End User

This level includes both the owner and those who make use of the data. The data owner uploads encrypted data and a list of all authorized users to the security environment.

(b) Security Level

The security level of the proposed system includes SM-PHC security framework, which operates using the following workflow:

1. File uploading: The user initiates the process by selecting a file from their local machine and uploading it through the interface of the security framework. This file is then stored in the local database for safekeeping.

2.File Encryption (Enc): The security framework employs Pallier Homomorphic Encryption (PHE) to convert the plain text (PT)of the uploaded file to cipher text (CT). This conversion process happens during the encryption phase, and the resultant cypher text is secure and unreadable to any unauthorized parties.

3.Secure File Uploading: The Enc file is then uploaded to the public cloud, where it is stored securely to ensure that no unauthorized persons can access it.

4.File downloading: A user submits the file name to the cloud in order to get a file. The user receives the file in an encrypted format after the cloud system looks for it.

5.File (Dec) decryption: The encrypted file is decrypted using the PHE decryption algorithm. This algorithm requires the private key that was used during the encryption phase to convert the CT back to PT.

6.Optimal Key generation: The security framework generates a pair of keys optimized by the spider monkey optimization method (SMO). These keys are used to ensure secure encryption and decryption of files, and the optimization process helps for Enc system optimization.

(c) Cloud storage level

A cloud storage is the highest level component of the security concept described in this study. The cloud interface is used by the security framework to upload encrypted user data, which is subsequently stored there. When necessary, users can download encrypted files using the security framework. Once the security framework has encrypted the file, it is sent to the user. Additionally, a multi-cloud environment can make use of this system concept. Nonetheless, the study's main focus is on the Spider Monkey (SM)-PHE technique. To provide the most effective solution for data security in CC, this technology encrypts and decrypts data before uploading it to the cloud.

3.2. Pallier Homomorphic Encryption (PHE)

Data encrypted with a CT can be processed using HE. This Enc method produces Enc results and enables computations on CT [23]. The computation outcome acquired in the encrypted form is Dec and compared with the PT version, as though the identical computation process was carried out on the PT. Figure 2 presents the processing flow in both Enc and unencrypted stages.

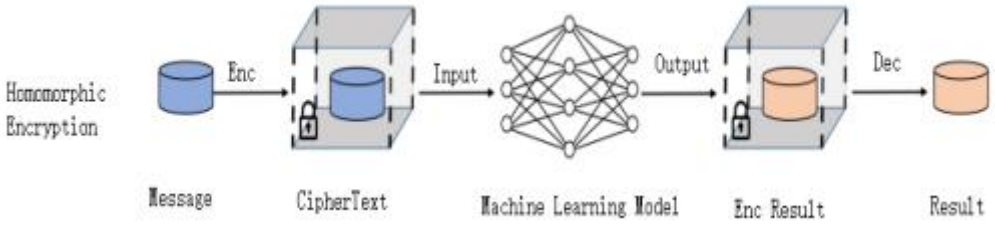

Figure 2. Process of PHE

HE is currently the most widely used PP mechanism since it may process CT without initially Dec it. In order to secure sensitive data and information during computing operations, the HE method can compute the CT without initially Dec it. This means that the computation party only has to receive the CT and not the contents of the PT. With the Enc content, HE can perform particular algebraic operations and process cryptographic data efficiently. Equation (1) illustrates that quaternions make up the HE cryptosystem.

$$H = \{Ho_{key}, Enc, Dec, Eval\} \quad (1)$$

Here, the key generation function is represented by Ho_{key}. The Enc function is denoted as Enc . The decryption function is represented by Dec, and the evaluation function is represented by $Eval$. A homomorphic operation that enables secure cryptosystems similar to Eq (2). The Enc that uses the public key (pub) as the Enc key is represented as $Enc_{pub}(.)$. The CT space is represented by C, and the PT space is by M.

$$\forall m_1, m_2 \in M, Enc_{pub}(m_1 \odot_M M_2) \leftarrow Enc_{pub}(m_1) \odot_c Enc_{pub}(m_2) \quad (2)$$

$\odot_M$ represents the operator on the M, and $\odot_c$ represents the operator on the C.The outcome is the same as if m_1 and m_2 were encrypted first and the operators were then run, as demonstrated by Eq. (2), for any two items m_1 and m_2 in the M, following the application of the $\odot_M$ operating on them. The absence of the intermediate decryption step and the equality or direct computation of the left-hand term from the right-hand term can be denoted by the symbol $\leftarrow$.

The result of HE for the PT v can be represented by [[v]], which simplifies the formula. The following defines addition HE and multiplication HE, the two fundamental HE operations.

Definition 1: Additive homomorphic operation.

Here, [[u]] and [[v]] represents the encryption outcomes for any 2 elements u and v in PT space, and if Eq (3) is met, Dec_{pri} represents the private key employed for decryption as:

$$Dec_{pri}([[u]] + [[v]]) = Dec_{pri}([[u + v]]) = u + v \quad (3)$$

Definition 2: Multiplicative homomorphic operation.

Here, [[u]] and [[v]] represents the encryption outcomes for any 2 elements u and v in PT space, and if Eq (4) is satisfied with Dec_{pri}:

The is represented as

$$Dec_{pri}([[u]] \times [[v]]) = Dec_{pri}([[u \times v]]) = u \times v \quad (4)$$

3.2. Paillier algorithm

The following are the procedures involved in creating the public-private key pair and the Enc and Dec principles in the Paillier algorithm.

Key generation: Initially, choose 2 huge prime numbers, a and b, at random. Make sure that a and b are the same length. Then, lcm is a function that determines the least common multiple and Proceed by computing n=ab and $\lambda = lcm(a-1, b-1)$. To satisfy Eq (5), randomly choose a positive integer g smaller than n^2 after defining $L(x) = \frac{x-1}{n}$.

$$\gcd\left(L\left(g^{\lambda} mod\ n^2\right), n\right) = 1, u = \left(L\left(g^{\lambda} mod\ n^2\right)\right)^{-1}, mod\ n\Big) \quad (5)$$

A function to determine the greatest common divisor is called gcd. This approach can be used to get the public key, which is known as the Pub Key (n,g), and the private key, which is known as the Secret Key (λ,u).

Enc procedure: Select a random number r such that 0<r< n for any m. Eq (6) determines the c.

$$c = g^m r^n mod n^2 \quad (6)$$

Dec procedure:

From Eq (7), the m for the c may be obtained.

$$m = L\left(c^{\lambda} mod n^2\right) * u\ mod\ n \quad (7)$$

The Paillier algorithm is an example of an asymmetric encryption mechanism that can be used to decrypt encrypted data and produce encrypted outcomes. The outcome that is reached is identical to the one that happens when the PT is operated on directly. Nevertheless, the multiplicative homomorphic operation is not satisfied by the Paillier algorithm. Despite not being fully HE, the Paillier algorithm is commonly employed in industry due to its excellent computational efficiency. The simulation algorithm of HE employed in this paper is the Paillier algorithm.

- **Calculation of the encryption loss function**

The public Keys and private keys in a ML model that protects the training parameters using the Paillier algorithm are typically generated randomly on the server. Data is primarily decrypted via secret key and encrypted using the Pub key. Typically, in ML models, an optimization technique like a Stochastic Gradient (SG) that minimizes the value L(θ;x)is applied once a loss function (L) has been created. The minimal value of L(θ;x) that is optimal is found using the parameter θ* descent.

Using (LR) Logistic Regression as an example, let $T = (x_1, y_1), (x_2, y_2), \ldots\ldots\ldots (x_n, y_n)$ represent the current collection of n sample data points. Then, in Eqn 8, the logarithmic L as its target L were implemented:

$$L = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i \theta^r x_i}\right) \quad (8)$$

By using the partial derivative θ from the previous Eq (8), the model parameters are updated. Equation (9) illustrates how to include the acquired gradient values into the gradient descent equation, thus the model parameters are updated.

$$\theta = \theta - \mathrm{lr} * \frac{\partial \mathrm{L}}{\partial \theta} \quad (9)$$

Until the value of the loss function L(;) and the computing process outlined above is repeated after x stops dropping or the maximum number of iterations is reached. The iteration is terminated at this point . There is a

chance of data leaking in the ML scenario, as the computing process described including the parameters and data being done in an explicit state. It is necessary for the parameters to be solved in the encrypted state for ML based on HE. Typically, the encrypted value [[θ]]is the transmitted parameter θ. In Eq. (10), the loss function is presented.

$$L = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i[[\theta]]^r x_i}\right) \qquad (10)$$

To calculate the L, complex exponential and logarithmic operations must be performed on the encrypted data since the Paillier algorithm only permits addition homomorphism and scalar multiplication homomorphism, not multiplication homomorphism or complex exponential and logarithmic operations. As a result, the encrypted form of the aforementioned Eq (10) cannot be solved.Here, the original logarithmic L can be approximated using the Taylor L by using the original logarithmic L's Taylor expansion. After Taylor expansion, the L is reduced to merely scalar multiplication and addition operations, allowing Paillier to be applied directly to the cryptographic solution. The logarithmic L is approximated by polynomials.

Many large prime power operations are required when utilizing the Paillier method for Enc and Dec. therefore intermediate outcomes typically lead to overflow problems and may be out of bounds. Consequently, when the number of local training iterations exceeds a predetermined number of rounds, build the FNN with HMDH algorithm to re-encrypt the data using the server-side key.

Like previous HE methods described in the literature, the PHE method introduces noise into CT throughout the Enc process. Encrypting the same PT with two distinct encryption step activations would result in two separate CT. As a result, this is necessary to guarantee the PHE scheme's Attribute Based Enc.The disadvantage is that noise is added in addition to being multiplied when performing homomorphic addition and multiplication on the CT.

In the event that one of the CT coefficients is rounded to the incorrect value throughout Dec (Eq. (4)), this could result in a critical situation. Thereby failing the step of Dec. One important component of the PHE design is noise handling. In the development of HE-based processing systems, an accurate assessment of the quantity (and kind) of operations permitted on the CT is essential. The noise budget (NB) is relevant in this particular situation. An intuitive definition of the NB is an indication of the number of operations that can be executed on a CT before its Dec fails, even though a formal definition is outside the scope of this work.

As a CT goes through the processing pipeline, its NB attribute changes. It is determined by the PHE scheme's parameters and expressed as a positive integer. Right after the Enc phase, the NB is first assigned to the CT. The amount of NB that is available in a newly encrypted CT will often rise when n is increased. In contrast, during homomorphic operations, maximizing p and q will result in a rise in NB usage. For HE-based systems, especially those that use DL methods, determining the values of _ that ensure the correct processing of the CT while lowering computation and memory complexity is essential.

The NB is reduced when HE operations: + and x are performed on the CT. It is important to emphasize that in order for the Dec process to operate properly, the CT must be decrypted before the NB decreases to 0. Although NB computation is quite complicated, there are specialized HE tools available for NB estimate.

3.3. Privacy preserved using hybrid Convolutional Neural Network (CNN) with Recurrent (NN) Neural Network (RNN) and (SSO) Swallow Swarm Optimization

The 2nd study's two primary contributions, describing in detail the process for creating PP CNNs using HE, is the objective of this section. It is necessary to reconsider and rebuild DL solutions in order to take into account the limitations on the kind and quantity of tasks that define the BFV system while creating PP DL solutions based on HE. The current study focuses on CNNs, which are the SOTA solution in various applications, among the many DL methods available.

When $l = 1, \ldots \ldots L$, a CNN F(.) is a deep NN with L processing layers $\eta_{\varepsilon l}^{(l)}$, each of which is defined by the parameters εl. In order to prevent the NB from running out while processing a PP CNN based on HE, the processing pipeline's length L must be carefully considered.

The processing layers in this CNN only consist of + and multiplication. An overview of the suggested process for creating PP CNNs based on HE is shown in Figure 2 [24].

The CNN that will be HE-encoded is indicated by F(.). By figuring out the configuration of the ⊖ encryption parameters, the methodology aims to produce a PP version' $\varphi \ominus (.)$ of F(.) and ensure that the NB does not exceed o during processing.

The methodology consists of three steps: model encoding, model validation, and model approximation, to accomplish these goals. The next sections provide more detail about these three steps.

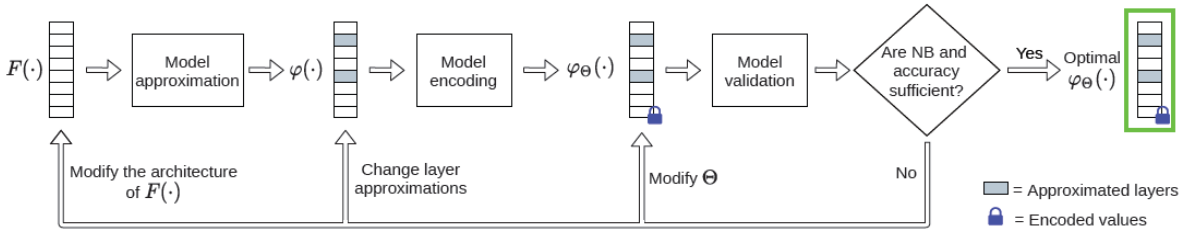

Fig. 2. An approach for the structure of PP CNNs depends on HE.

A. Model approximation (MA)

Its objective is to substitute processing layers that only perform adds and multiplications for those that do not adhere to the BFV scheme for F(.). In a DL scenario, where processing layers consist of division, square root, and nonlinear (AF) Activation Functions, this is crucial. An approximated model φ (.) with HE-compliant processing layers is the outcome of this stage. More specifically, the relevant approximated model φ (.) is output by the model approximation stage, which takes a CNN F(.) as input.

Each processing layer_(l)^_l in such a model is consistent with the BFV scheme since it only consists of additions and multiplications.

After this approximation step, the estimated CNN model has to be retrained.

1) **Pooling layers:** BFV scheme does not include the comparison operator, is used by maximum pooling layers. Different pooling methods might be employed in place of maximum pooling to solve this issue. Since the BVF scheme's average pooling only needs to be multiplied by the sum of the CT and a fixed value that is known in advance (i.e., $\frac{1}{k_w \times k_h}$). Here,the width of the pooling kernel is denoted as k_w and the height of the pooling kernel is denoted as $k_h$the authors suggest to replacing it with this method.

2) Normalization layers: Since it is difficult to calculate the mean and Standard Deviation (SD) of encrypted data, NL cannot be taken into account in the BFV scheme. On the other hand, because batch normalization layers rely on the values of the training data, they are accessible. These values can be utilized while processing CTand are calculated throughout the training phase.

3) Activation functions:

Nonlinear functions are commonly employed as AF in CNNs. Because it needs to apply the comparison operator, the ReLU AF cannot be computed. This also applies to the division-based hyperbolic tangent tanh. This work proposes to substitute the square AF $f(x) = x^2$ for these nonlinear AF. Taylor polynomial expansions can be used to improve this approximation even more. Still, utilizing more polynomials to make the expansion more accurate means that there will be more operations involved, which will increase the amount of NB used.

B. Model encoding (ME)

It can be encoded using the ME step Once the model is approximated. This produces an encoded approximation model $\varphi \ominus (.)$ with weights encoded based on the parameters ⊖ and the PHE scheme. $\varphi \ominus (.)$ may now process encrypted data and return the CNN processing result. Interestingly, this result is still encrypted, meaning that it can only be decrypted by the owner of the secret key, sk. It is still an interpretation how the ideal ⊖ parameter setting serves for HE processing. Usually, a "trial-and-error" method is used to choose the values for these parameters. However, the following instructions are offered for the setting of ⊖.

The most important encryption parameter is n, since it affects both the computing expense of the encrypted processing and the initial NB setting. NB adequate values are normally only guaranteed for extremely simple machine learning (ML) models (usually consisting of two or three processing layers at most) when n is smaller than 4096. Generally, n is first set at 4096 and then increased from a methodological perspective. In other words, the parameter p influences the likelihood that some coefficients of the decrypted polynomials will be rounded to the incorrect value, which in turn impacts both the precision of the homomorphic operations and the quantity of NB spent.

A number of p between 2^{16} and 2^{18} usually indicates an acceptable location to start when experimenting with the settings; tuning p is a process that involves trial and error. The value of q is crucial to the scheme's security; to set q in accordance with n and p, it is advised to rely on the aid function offered by SEAL. The DL model $\varphi \ominus (.)$ can be obtained by encoding $\varphi(.)$, with the obtained $\ominus = (n, p, q)$.

C. Model validation (MV)

Following the completion of the encoding process, the encoded model $\varphi \ominus (.)$ is assessed from two distinct angles by means of this stage. In order to ascertain whether the selected configuration $\ominus$ provides a strong enough NB during CT processing, $\varphi \ominus (.)$ is initially looked at.

The accuracy loss of $\varphi \ominus (.)$ with respect to F(.) is assessed in the second place $\varphi \ominus (.)$processes a (potentially huge) set of raw messages ms in order to measure the NB of the final CT . For attaining these goals, assessing the difference in accuracy among the encoded model $\varphi \ominus (.)$and that of the plain model F(.).

The difference among the accuracy of $\varphi \ominus (.)$ and F(.) and final CT NB can be assessed. The incorrect setting of $\ominus$ is the basis for the issues related to NB loss. But, the approximations of processing layers that are presented in MA step and in $\varphi \ominus (.)$, encoded approximated model, p and q are too small for pipeline processing, these 2 will be related to the discrepancies in the output among $\varphi \ominus (.)$ and F(.).

Generally, the issues related to a loss of NB are dependent on an incorrect setting of $\ominus$. If the NB constraint is met and the accuracy loss is less than a user-specified threshold (like 1% or 5%), then $\varphi \ominus (.)$ and that of the F(.). The PP variant of F(.) to be taken into consideration is usually the one whose issues stem from a loss of NB due to an improper setup.

Alternatively, when both the plain model F(.) and the NB of the CT decline to 0 during the processing of $\varphi(.)$the NB condition is not satisfied.

The methodology recommends 3 diverse tasks: update $\ominus$, change the manner in which layers in F(.)change the processing pipeline of F(.) or approximated. An incorrect setup or a loss of accuracy greater than the threshold typically causes the problems connected with a loss of NB.

The following is a detailed description of these three actions.

Initially, $\ominus$ is the only factor that determines the NB and accuracy loss. Specifically, raising the parameter n results in an initial NB increase, but at the rate of an upsurge in computational overhead (which could be significant) and memory consumption of $\varphi \ominus (.)$. On the other hand, raising p and q would result in higher NB consumption by the HE processes but less accuracy loss (by improving processing precision). Second, for the processing layers in F(.) that do not comply with HE, alternative model approximations could be taken into consideration.

In this case, a trade-off needs to be thoroughly considered. In fact, a finer approximation for the (CG) coarse-grain layer could be used to reduce the accuracy loss (e.g., greater degree of polynomial approximation was employed); this would further reduce the NB by requiring more operations to be completed for that layer. Conversely, switching from a fine-grain to a CG layer approximation may lessen the amount of NB used, but it may also result in a greater accuracy loss.

Third, a updated form F'(.) of F(.) can be constructed if the first two steps are unsuccessful in meeting the requirements on NB and accuracy. This objective of lowering the number of operations to be done can be attained in two ways: through lowering the amount of processing layers or by simplified the operations that must be performed.

The stages of model approximation, encoding, and validation are reactivated to identify $\varphi \ominus (.)$ once F'(.) has been rebuilt.

3.3.1. CNN combined with RNN

Data representations at different levels of complexity can be obtained using computational models made up of many processing layers via DL. These representations are then used to make predictions. To improve PP performance more accurately, the notion of hybrid learning is presented in this work. The RNN and the CNN are integrated.

One framework that has been suggested by the advancement of biotechnology is the CNN. Together, neurons are arranged in a well-organized manner, similar to local filtering of the whole input space. The deep features and local features of the input data can be extracted by CNN [26]. Neural networks that process sequence data are called RNN. Each layer and the nodes that connect it are connected, beginning with the input layer (IL) and continuing through the hidden layer (HL) and output layer (OL) in the CNN structure.

Such sequential data cannot be handled by such a network approach. In order to create higher order features, the CNN layer first learns low-level translation invariant features, which it then feeds into multiple fixed-tree RNNs.

Convolution Layer (CL) and pooling layer can be considered as being combined into one effective, hierarchical process by RNNs. When compared to a-priori techniques, these two models both produce superior outcomes. Combining the usage of CNN and RNN frameworks for the classification of blood cell images was inspired by these works. Furthermore, the CNN-RNN model is the one put forth in this paper. A training phase and a testing phase are part of the suggested methodology. Pre-trained the CNN model on the dataset was the initial step in the training process. After that, a new CNN is initialized using pre-trained network parameters with the use of a Transfer Learning (TL) technique. After that, the RNN model is trained and all CNN layers are blocked. Both features generated from the CNN and RNN are combined simultaneously using NN attention processes. During the testing phase, the refined CNN-RNN model receives the pre-processed test data, and the Softmax layer is used to extract the classification results. Below is a description of further information.

The pre-trained CNN layer, RNN layer, merge layer, and Fully Connected (FC) layer with Softmax output make up this suggested model's component parts.

1) Pre-Trained CNN Layer

As the CNN model's initialization weights, through pre-trained on dataset, the weight parameters that have been obtained. CNN consists of pooling layers and CL.

2) CL

Applying convolution operations on the Feature Map (FM) of the preceding layer using convolution windows of different sizes, this layer is the most crucial and it is computed in the CNN. Various-sized convolution windows slide sequentially onto the previous layer's FM [25]. Typically, the CL number of weight parameters varies with the Window Size (WS), which is either 3x3 or 5x5. Each feature map's neuron values in the CL are convoluted through corresponding windows, and the AF employed in the layer determines the final outcome.

3) Pool Layer

This layer calculates in a manner similar to that of the CL. The difference is because the lowest sample layer's (SW) sliding window tends to be 2 x 2, whereas the sliding step is usually 2. As a result, this procedure will typically cut the FM size from the preceding layer by half. This can significantly lower the convolution weights of NN parameters, it facilitates the overall efficiency of the network training process. It also makes it possible for the network to adjust to variations in image scale more effectively. Utilize the ReLU (Linear Rectification Function) as the AF in this study.

4) RNN Layer

There are three layers in RNN and CNN: IL, HL, and OL. The most crucial aspect of RNN is the manner in which these HL are connected [26]. The HL is output to the OL, and the IL and HL nodes are coupled to one another. The HL node receives the node output data back. Data regarding nodes that are adjacent to one another in

the HL might even be contained. This network is dynamic. RNNs are more closely related to the biological nervous system since biological NN are cyclic networks with the ability to comprehend serial data. Long-term dependency information can be learned via this method.

The cell is the name of this processor's structural component. The three gates that exist in a cell are an input gate (IG), an output gate (OG), and a forget gate (FG).

Rules can be used to evaluate a message once it enters the LSTM network. The Oblivion Gate will remove the contradictory data, leaving only the data that satisfies the standards for validation of the algorithm. The IL, OL, and FG are the three multiplicative gates that make up an LSTM, along with a memory cell. Compared to a traditional RNN repeat module, the LSTM has a more complex internal structure.

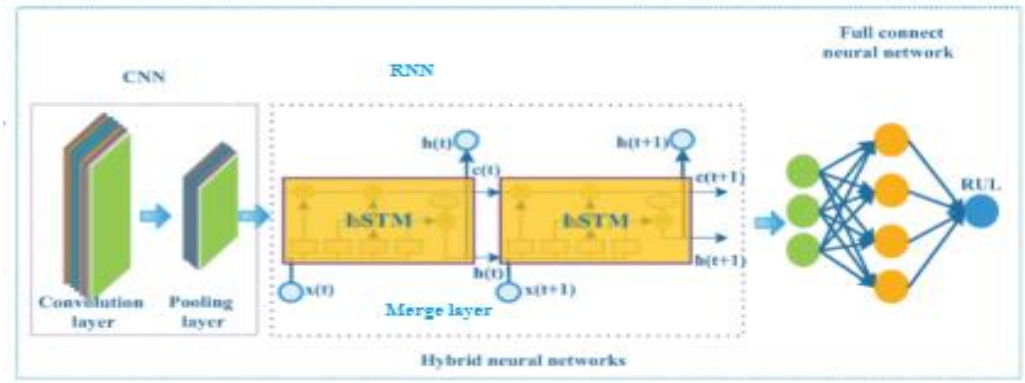

Figure 3. Suggested Hybrid DL framework

5) Merge Layer

The purpose of this layer is to combine Features Extracted (FE) from the RNN and the CNN using a particular technique. Introduce NN Attentional Mechanisms (AM) in the sequential model. With the help of its AM, a NN can selectively choose and focus on particular input features and for feature fusion, use the corresponding (EWM) Element-Wise Multiplication procedures.

6) FC Layer with Softmax Output

The output of the FC Softmax layer, which receives the FE from the combined RNN and CNN, is the probability distribution of all classes. Additionally, use the Cross-Entropy (CE) as a (LF) Loss Function to assess the difference between the desired and actual outputs.

7) Training of Network

This model has two separate branches. The RNN branch initializes the parameters at random, while the CNN branch uses weights based on pre-trained parameters from the dataset. The gradient of the CE LF is used to iteratively update these weights throughout the training phase. The RMS Prop optimizer calculates the training samples and needs 100 epochs to complete the training process after initially blocking the CNN layer. The CNN layer is then thawed the network as a whole computes training samples using the Adam optimizer, the learning rate is 0.0001, and training now takes 70 epochs. After a set number of intervals, the training procedure ends.

To learn compact representations, these combined models do not make use of time dependencies in features. Even with fewer samples available, deep architectures was trained due to these underlying time dependencies. Time dependencies frequently occur to improve time sequence modeling and time utilization. Therefore, in order to train the model in the shortest amount of time, this work used SSO to limit error propagation during training.

3.3.2. Swallow Swarm (SS) optimization (SSO)

The main inspiration for this novel optimization approach comes from SS.

In this algorithm, 3 kinds of particles are employed:

1. Explorer particle (e_i)

2. Aimless particle (oi)

3. Leader particle (li)

These particles constantly interact with one another as they travel in parallel directions. Every particle within the colony (which may consist of several subcolonies) is accountable for something, and by doing so, they help the colony move toward a more favorable state.

a) Explorer particle

These particles are primarily representative of the colony's population.Exploring problem space is considered to be the main task.Once the group reaches the extreme point (swallow), a unique sound is used to bring them there. If this location proves to be the best in the problem space, the particle assumes the position of a Head Leader (HLi) [27]. On the other hand, each particle e_i with respect to VHLi (velocity vector of particle toward HL), VLLi (velocity vector of particle toward LL), and competence of resultant of these two vector makes a random move. If the particle is in a good (though not the best) situation compared to its neighboring particles, it is chosen as a local leader LLi. The movement of a particle in problem space is seen in Figure 3.

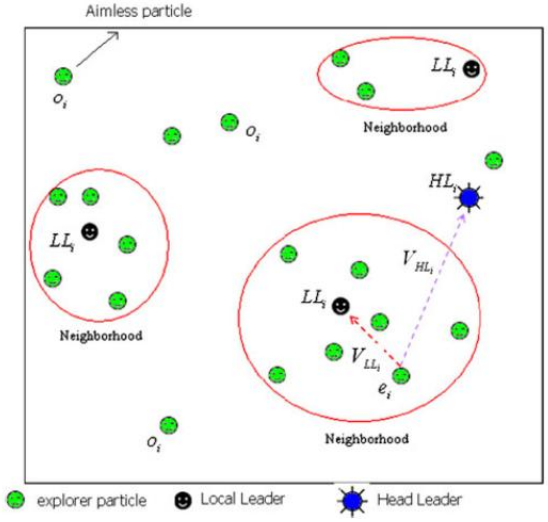

Fig. 3 Particle types and the movement of explorer particles

$$V_{HL_{i+1}} = V_{HL_i} + \alpha_{HL} rand()(e_{best} - e_i) + \beta_{HL} rand()(HL_i - e_i) \quad (11)$$

$$\alpha_{HL} = \{if(e_i = 0 || e_{best} = 0) \rightarrow 1.5\} \quad (12)$$

$$\alpha_{HL} = \begin{cases} if(e_i < e_{best})(e_i < HL_i) \rightarrow \frac{rand().e_i}{e_i.e_{best}} \\ if(e_i < e_{best})(e_i > HL_i) \rightarrow \frac{2rand().e_{best}}{1/(2e_i)} \\ if(e_i > e_{best}) \rightarrow \frac{e_{best}}{1/(2.rand())} \end{cases} \quad (13)$$

$$\beta_{HL} = \{if(e_i = 0 || e_{best} = 0) \rightarrow 1.5\} \quad (14)$$

$$\beta_{HL} = \begin{cases} if(e_i < e_{best})(e_i < HL_i) \rightarrow \frac{rand().e_i}{e_i.HL_i} \\ if(e_i < e_{best})(e_i > HL_i) \rightarrow \frac{2rand().HL_i}{1/(2e_i)} \\ if(e_i > e_{best}) \rightarrow \frac{HL_i}{1/(2.rand())} \end{cases} \quad (15)$$

The behavior of the e_i is considerably impacted by VHL_i. The particle's present location in problem space is denoted by e_i. The best position that the particle can recall from the start to the present is denoted by e_{best}. A leader particle with the best reaction at its current position is denoted by HL_i. The control of adaptively specified acceleration coefficients is given by α_{HL} and β_{HL}. The position of the particle affects how these two parameters vary as the particle moves. The chance of the particle being a global minimum should be taken into account, and control

coefficients should estimate a little amount to reduce the particle movement to the least, if the particle is a minimum point (minimizing problem) and is in a better position than the e_{best} and HL_i.

The particle ought to move toward $HL_i$with an average amount if it is in a worse state than HL_i but better than e_{best}. The particle can travel closer to $HL_i$with a greater amount if its location is poorer than e_{best}, which also makes it worse than HL_i. Consider that this movement is influenced by the VLL_ivector.

$$V_{LL_{i+1}} = V_{LL_i} + \alpha_{LL} rand(\,)(e_{best} - e_i) + \beta_{LL} rand(\,)(LL_i - e_i) \quad (16)$$

$$\alpha_{LL} = \{if(e_i = 0 || e_{best} = 0) \rightarrow 2\} \quad (17)$$

$$\alpha_{LL} = \begin{cases} if(e_i < e_{best})\,(e_i < LL_i) \rightarrow \frac{rand(\,).e_i}{e_i.e_{best}} \\ if(e_i < e_{best})\,(e_i > LL_i) \rightarrow \frac{2rand(\,).e_{best}}{1/(2e_i)} \\ if(e_i > e_{best}) \rightarrow \frac{e_{best}}{1/(2.rand(\,))} \end{cases} \quad (18)$$

$$V_{i+1} = V_{HLi+1} + V_{LLi+1} \quad (19)$$

$$e_{i+1} = e_i + V_{i+1} \quad (20)$$

To calculate the vector of $VLLi$, each particle e_i makes use of the nearest particle, LLi.

b) Aimless particle (o_i)

In the early stages of ei, these particles have a poor amount of f(o_i) and a worst position relative to other particles. A new group reponsibility for them (o_i) is defined after identifying that these particles can be distinguished from explorer particles ei. Exploratory and random search are included in this task. They move at random and are unaffected by HLi and LLi positions when they begin. These swallows serve as the colony's explorers, venturing into isolated regions and reporting significant results to the rest of the group.

The group converges to a local optimum in many optimization problems because of the improper distribution of particles in position space, which hides the optimal solution. Early convergence in local optimum pints is the most difficult problem to solve when dealing with optimization problems. Although (o_i) appear to be aimless and useless, consider the possibility that they will ignore the global optimum solution, avoid the many surrounding points with their long hops, and consider the optimization problem. The local optimum locations, LLi and HLi, are compared by the particle oi with its current position. This particle will switch its position with the closest explorer particle, e_i, if it reaches an optimal point while searching, and it will then resume its search.

$$o_{i+1} = o_i + \left[rand(\{-1,1\}) * \frac{rand(min_s, max_s)}{1+rand(\,)}\right] \quad (21)$$

Each o_i particle's new position is equal to its current position with a random number among the position space's lowest and maximum, divided by a value between one and two. Particle o_i is randomly moved to or from its prior position based on the division answer. (-50, 50) is used to define the function Rosenbrock's range. The fraction result would be 12.5 if the function rand (min, max) yields a random number of 25, and the function *rand*() provides 0.5. At this point, the value of oi may be increased or decreased by this number. Examining the many environmental areas will be more likely increased as a result.

c) **Leader particle**

Leader particles are part of the SSO algorithm. When position space searching first starts, these particles have the optimal value of f(Li). In each level, their location and number may vary. While the new approach may contain nl leader particles, the PSO method only contains one (gbest). In space, these particles could be dispersed or accumulated. There are some particles called LLi, and the greatest leader is called Leader Head, which is acknowledged as the colony's main leader. The responses we keep find them to be excellent candidates. A thousand-member colony of swallows is subdivided into several sub-colonies in the real world.

There is a LLi in all of these subcolonies, but other swallows who are stronger and smarter may frequently replace them. A bird that is in a superior location, close to food and a resting area, is considered the leader of the swallow population. Guiding other colony members to this location is the leader's responsibility. The SSO algorithm

simulates this problem. Every time an issue is repeated, either a head or Li acting as a leader may change, or $o_i$may find a solution that works best at that particular location thus far. Since swallow movements occur quickly and dynamically, actual borders between subcolonies might never be created.

The number and distribution of swallows in the diagram varies with the size of subcolonies. Each particle, swallow, is capable of fulfilling one of these 3 tasks. These three particles, e_i, o_i, and Li, interact with one another on a constant basis. When these particles may often switch roles during the search phase, identifying the optimal position is the more crucial task.

4. Results and Discussion

Initially, a medical diabetic dataset is obtained from the internet and used to train the system using 800 samples and the key is generated using paillier homomorphic encryption (PHE), which can be optimized using optimization. As a result, which gives optimal keys for encryption and decryption. Finally, the selected dataset is encrypted using a public key, which is then stored securely in real time database of public cloud for further accessing. To maintain confidentiality and to prevent unauthorized access by third parties, the data is provided in an encrypted form when accessed from the public cloud by the user and data owner. The proposed mechanism established in this study provides protection against malicious activity and unauthorized access through the use of encrypted data. Performance metrics are compared with those of other current methods to evaluate the effectiveness of the suggested procedure. This section presents a comparison between the proposed technique and several existing homomorphic encryption methods, Key Homomorphic Encryption (KHE), Energy Efficient Dynamic Homomorphic Security (EE-DHS) spider-monkey wıth Paillier homomorphic encryption model (SM-PHE), and proposed Hybrid Fuzzy Neural Network with Modified Deer Hunt optimization based Pallier Homomorphic Encryption (HFNNMDH-PHE).

Encryption time: To calculate the encryption time, you can measure the amount of time required by the encryption algorithm to encrypt the plaintext into ciphertext. Generally, the encryption time can be calculated by subtracting the start time from the end time of the encryption process. Encryption time is a critical factor in evaluating the efficiency of an encryption algorithm, as it reflects the amount of time required to convert plaintext to ciphertext. The shorter the encryption time, the more efficient the algorithm. This Section compares the encryption time of proposed with that of other homomorphic encryption algorithms.

Table 1. Encryption time vs methods

Data Size (kb)	KHE	EE-DHS	SM-PHE	HFNNMDH-PHE	HCRNN-SSO
100	12	15	18	22	24
200	15	16	21	25	27
300	18	17	25	28	29
400	20	18	28	31	33
500	23	21	31	33	35

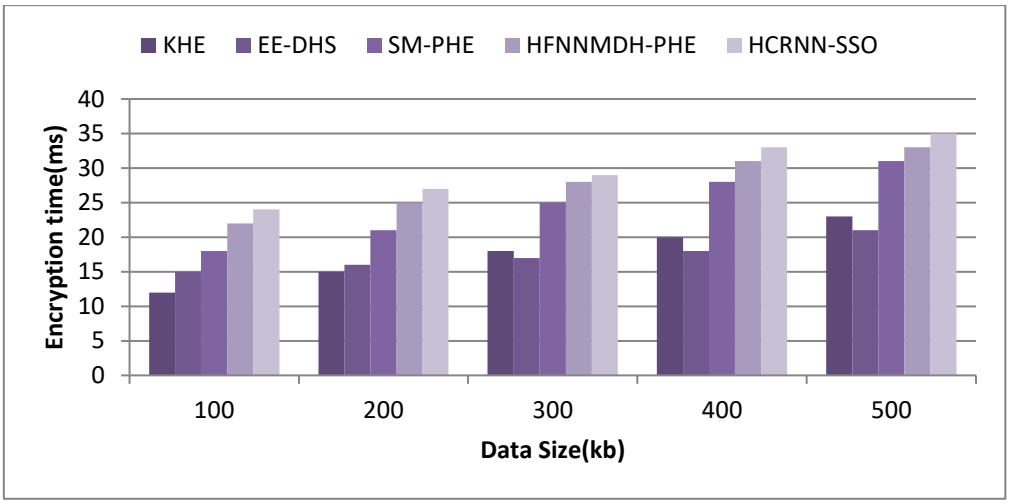

Figure 5. Encryption Time

The results of figure1. demonstrate that proposed model takes less time to encrypt the same plaintext compared to the other algorithms, indicating its superior efficiency in this aspect. The intended HFNNMDH-PHE model achieved 24 ms for 100kb using this metric. As a result, the EE-DHS and SM-PHE methods achieved 27, and 29 ms, respectively, while the KHE approach reached 33 ms for 100kb, as shown in Fig 5.

Decryption Time: The length of time required to decrypt encrypted data or information is referred to as decryption time. Decryption is the process of turning encrypted or encoded material back to its original form so that the intended recipient can read and understand it. The time taken to decrypt data based on the encryption algorithm employed, the size of the data being decrypted, and the computational capability of the device used for decryption. In general, more advancedencryption algorithms and higher data volumes will necessitate more decoding time. Using this criteria, the proposed model accomplished 5 ms for 100kb**.**

Table 2. Decryption time vs methods

Data Size (kb)	KHE	EE-DHS	SM-PHE	HFNNMDH-PHE	HCRNN-SSO
100	13	11	9	6	4
200	17	14	10	7	6
300	22	19	15	10	8
400	25	21	18	13	10
500	29	25	21	15	12

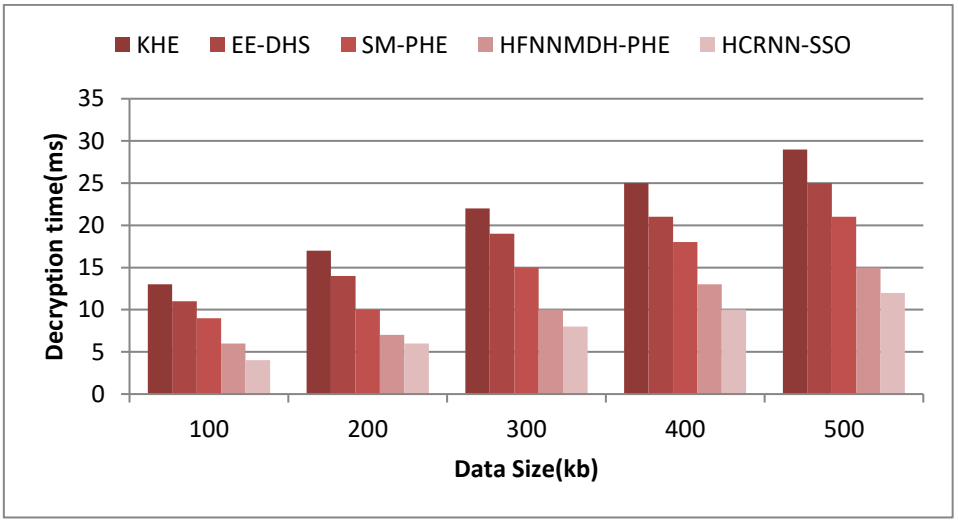

Figure 6. decryption time

As a result, for data sizes of 200 kb, 300 kb, 400 kb, and 500 kb, the decryption time is 4, 6, 10, and 12 milliseconds.The proposed HFNNMDH-PHE technique gained less decryption time when compared to other models expressed in fig 6.

Execution Time: The execution time of an algorithm is the amount of time it takes for the algorithm to complete its task or solve a problem. Execution time is typically measured in units of time, such as seconds, milliseconds, or microseconds.

Table 3. Execution time vs methods

Data Size (kb)	KHE	EE-DHS	SM-PHE	HFNNMDH-PHE	HCRNN-SSO
100	0.3	0.28	0.24	0.06	0.04
200	0.34	0.32	0.2	0.08	0.06
300	0.38	0.36	0.18	0.1	0.1
400	0.42	0.4	0.16	0.12	0.14
500	0.46	0.45	0.18	0.14	0.13

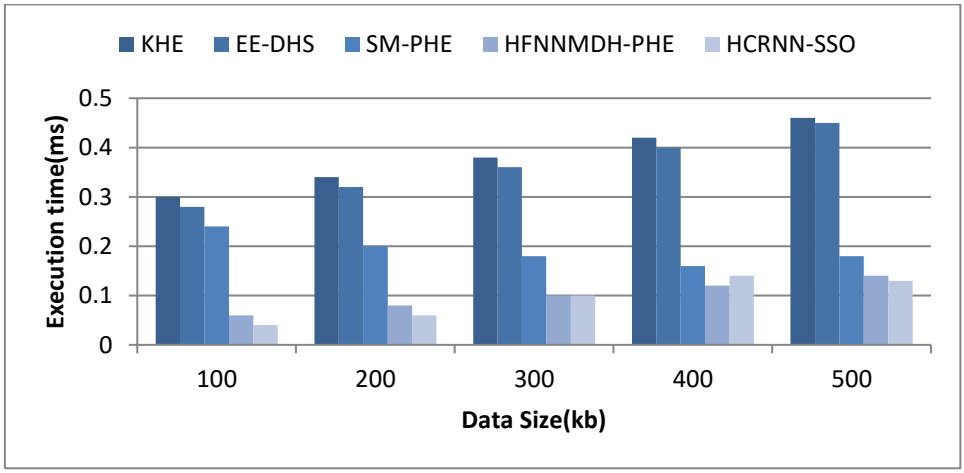

Figure 7. Execution time

The specific unit used may depend on the granularity required to accurately measure the execution time of the operations, which can be expressed in fig 7. The proposed model takes the less time for the execution compared with the other models.The proposed model achieves shorter execution time of 0.04 ms for 100 kB, 0.06 ms for 200 kB, 0.10 ms for 300 kB, 0.14 ms for 400 kB, and 0.13 ms for 500 kB of data size as a result in fig 3.

Efficiency: The efficiency of an algorithm can be evaluated based on several factors, including computational complexity, memory requirements, power consumption, and communication overhead. These factors determine how quickly and efficiently the algorithm can perform its cryptographic operations while using the minimum possible resources and proposed model is more efficient than other homomorphic techniques.

Table 4. efficiency vs methods

Data Size (kb)	KHE	EE-DHS	SM-PHE	HFNNMDH-PHE	HCRNN-SSO
100	92.67	91.45	89	98.5	99
200	91.64	90.85	90.36	98.2	98.7
300	92.68	91.36	91.5	98	98.4
400	93.47	92.35	92.12	97.9	98.3
500	94.57	92.68	91.55	97.7	98.8

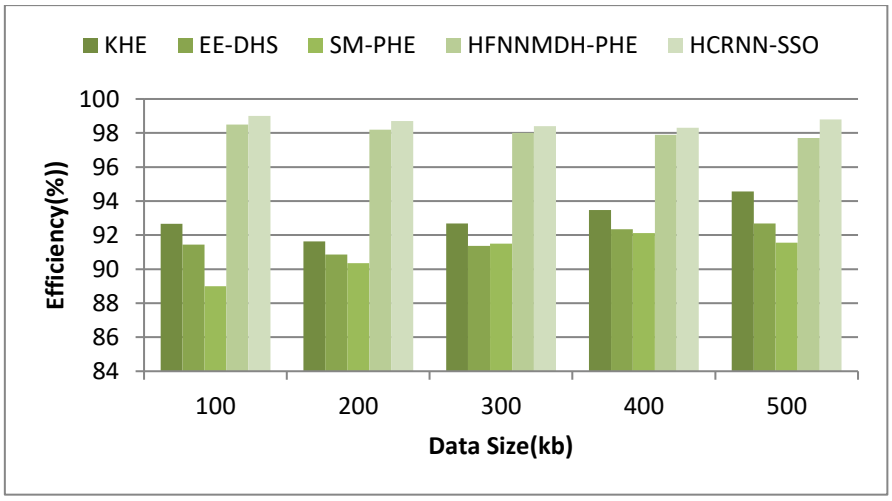

Figure 8. efficiency

In this case, the suggested model's efficiency was superior when compared to other frameworks, as a consequence, proposed methods effectiveness is > the standard methods in the following cases: 99% for 100 kb, 98.7% for 200 kb, 98.4% for 300 kb, 98.3% for 400 kb, and 98.8% for 500 kb in the fig 8.

Power consumption: The power consumption can be estimated based on several factors, including the size of the public key, the length of the plaintext and ciphertext, and the number of operations required to perform the homomorphic computation. The less power consumption leads to secure and energy-efficient system.

Table 5. Power consumption vs methods

Data Size (kb)	KHE	EE-DHS	SM-PHE	HFNNMDH-PHE	HCRNN-SSO
100	28	25	30	15	13
200	30	28	32	18	15
300	32	30	35	20	17
400	34	32	38	22	19
500	35	34	40	25	20

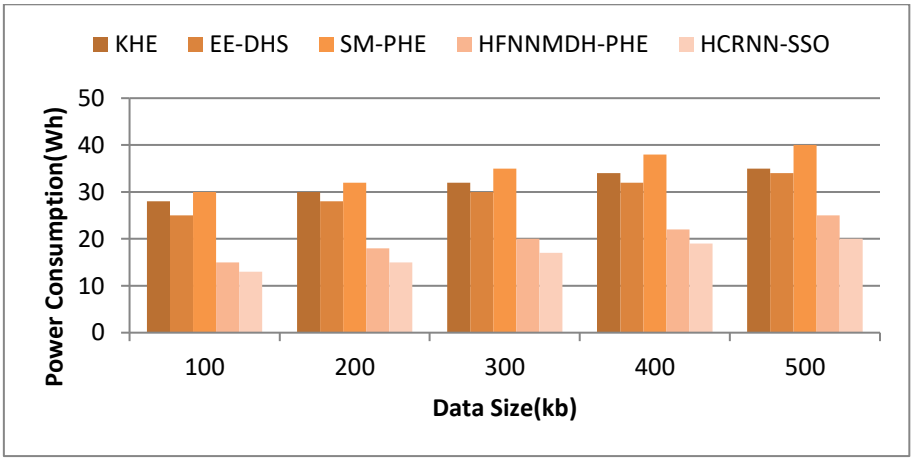

Figure 9. power consumption

The suggested model uses less energy than the current models, such as HCRNN-SSO, which uses 13Wh for 100kb of data and 15Wh, 17Wh, 19Wh, and 20Wh for 200kb, 300kb, 400kb, and 500kb, respectively in fig.9.

5. Conclusion

There's a need to use security algorithms more frequently in data systems and processes due to increased awareness and concerns about CC, information security, and privacy protection. Based on the communication mode, this research created a MD security sharing scheme and built a DL framework that protects training parameters using HE technology. In this case, the training parameters are protected by using the PHE to achieve additive homomorphism. This article offers a PP hybrid CRNN based on SSO to address the issue of privacy leaking. By merging DL with HE, a knowledge transfer strategy with PP is created. The suggested method performs an effective task in securing the data owner's privacy. In order to compare the efficiency of the suggested scheme with the state-of-the-art plan, the final result is a detailed evaluation. The outcome demonstrates that, in comparison to alternative methods, the suggested plan offers efficiency gains of up to 99%. This increase has demonstrated that the suggested strategy is better suited to run the data collecting method in a mobile CC environment.

REFERENCES

[1] Balasubramaniam, S., & Kavitha, V. (2013). A survey on data retrieval techniques in cloud computing. *Journal of Convergence Information Technology*, *8*(16), 15.

[2] Dawoud, M., & Altilar, D. T. (2016, April). Privacy-preserving Data Retrieval using Anonymous Query Authentication in Data Cloud Services. In *CLOSER (2)* (pp. 171-180).

[3] Li, J., Lin, D., Squicciarini, A. C., Li, J., & Jia, C. (2015). Towards privacy-preserving storage and retrieval in multiple clouds. *IEEE Transactions on Cloud Computing*, *5*(3), 499-509.

[4] Song, W., Wang, B., Wang, Q., Peng, Z., Lou, W., & Cui, Y. (2017). A privacy-preserved full-text retrieval algorithm over encrypted data for cloud storage applications. *Journal of Parallel and Distributed Computing*, *99*, 14-27.

[5] Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., & Buyya, R. (2016). Ensuring security and privacy preservation for cloud data services. *ACM Computing Surveys (CSUR)*, *49*(1), 1-39.

[6] Bülbül, B., & Altılar, D. T. (2019, September). Privacy preserving data retrieval on data clouds with fully homomorphic encryption. In *2019 4th International Conference on Computer Science and Engineering (UBMK)* (pp. 1-6). IEEE.

[7] Xia, Z., Zhu, Y., Sun, X., Qin, Z., & Ren, K. (2015). Towards privacy-preserving content-based image retrieval in cloud computing. *IEEE Transactions on Cloud Computing*, *6*(1), 276-286.

[8] Mittal, S., Ramkumar, K. R., & Kaur, A. (2021, October). Preserving privacy in clouds using fully homomorphic encryption. In *2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)* (pp. 1-7). IEEE.

[9] Hariss, K., Chamoun, M., & Samhat, A. E. (2020, October). Cloud assisted privacy preserving using homomorphic encryption. In *2020 4th Cyber Security in Networking Conference (CSNet)* (pp. 1-8). IEEE.

[10] Wang, X., Luo, T., & Li, J. (2020). An efficient fully homomorphic encryption scheme for private information retrieval in the cloud. *International Journal of Pattern Recognition and Artificial Intelligence*, *34*(04), 2055008.

[11] Oh, E. N., Baharon, M. R., Yassin, S. M. W. M. S. M. M., Idris, A., & MacDermott, A. (2022, August). Preserving Data Privacy in Mobile Cloud Computing using Enhanced Homomorphic Encryption Scheme. In *Journal of Physics: Conference Series* (Vol. 2319, No. 1, p. 012024). IOP Publishing.

[12] Song, W., Peng, Z., Wang, Q., Cheng, F., Wu, X., & Cui, Y. (2014). Efficient privacy-preserved data query over ciphertext in cloud computing. *Security and Communication Networks*, *7*(6), 1049-1065.

[13] Kirubakaran, S. S., Arunachalam, V. P., Karthik, S., & Kannan, S. (2023). Towards Developing Privacy-Preserved Data Security Approach (PP-DSA) in Cloud Computing Environment. *Computer Systems Science & Engineering*, *44*(3).

[14] Yang, J. J., Li, J. Q., & Niu, Y. (2015). A hybrid solution for privacy preserving medical data sharing in the cloud environment. *Future Generation computer systems*, *43*, 74-86.

[15] Gajmal, Y. M., & Udayakumar, R. (2022). Privacy and utility-assisted data protection strategy for secure data sharing and retrieval in cloud system. *Information Security Journal: A Global Perspective*, *31*(4), 451-465.

[16] Wang, N., Zhang, S., Zhang, Z., Fu, J., Liu, J., & Wang, R. (2022). Block-based privacy-preserving healthcare data ranked retrieval in encrypted cloud file systems. *IEEE Journal of Biomedical and Health Informatics*, *27*(2), 732-743.

[17] Li, J., Lin, D., Squicciarini, A. C., Li, J., & Jia, C. (2015). Towards privacy-preserving storage and retrieval in multiple clouds. *IEEE Transactions on Cloud Computing*, *5*(3), 499-509.

[18] Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., & Buyya, R. (2016). Ensuring security and privacy preservation for cloud data services. *ACM Computing Surveys (CSUR)*, *49*(1), 1-39.

[19] Wang, C., Cao, N., Li, J., Ren, K., & Lou, W. (2010, June). Secure ranked keyword search over encrypted cloud data. In *2010 IEEE 30th international conference on distributed computing systems* (pp. 253-262). IEEE.

[20] Cao, N., Wang, C., Li, M., Ren, K., & Lou, W. (2013). Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on parallel and distributed systems*, *25*(1), 222-233.

[21] Liang, K., Huang, X., Guo, F., & Liu, J. K. (2016). Privacy-preserving and regular language search over encrypted cloud data. *IEEE Transactions on Information Forensics and Security*, *11*(10), 2365-2376.

[22] Xia, Z., Wang, X., Sun, X., & Wang, Q. (2015). A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE transactions on parallel and distributed systems*, *27*(2), 340-352.

[23] Fazio, N., Gennaro, R., Jafarikhah, T., & Skeith, W. E. (2017). Homomorphic secret sharing from paillier encryption. In *Provable Security: 11th International Conference, ProvSec 2017, Xi'an, China, October 23-25, 2017, Proceedings 11* (pp. 381-399). Springer International Publishing.

[24] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, *77*, 354-377.

[25] O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

[26] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*.

[27] Neshat, M., Sepidnam, G., & Sargolzaei, M. (2013). Swallow swarm optimization algorithm: a new method to optimization. *Neural Computing and Applications*, *23*(2), 429-454.