UtilitasMathematica

ISSN 0315-3681 Volume 122, 2025

Using Multi-Level Federated and Lightweight Deep Learning Assisted Homomorphic Encryption Based on AI Technology to Provide End-to-End Security and Privacy Preservation for Multi-Cloud Environments

Adebanjo Ambrose Falade[1], Gaurav Agarwal[2*], Akash Sanghi[3]

[1]Research Scholar, Department of CSE, Invertis University, Bareilly, Uttar Pradesh-243123, India.

[2*]Associate Professor, Department of CSE, Invertis University, Bareilly, Uttar Pradesh- 243123, India.

[3]Associate Professor, Department of CSE, Invertis University, Bareilly, Uttar Pradesh- 243123, India.

E-mail ID's: - faladeadebanjo@gmail.com, gaurav.a1@invertis.org*, sanghiakash@gmail.com.

Abstract: As cloud computing (CC) expands, security and privacy become more important. The main problems of increasing processing overheads, communication bottlenecks, and noise budgets are addressed by this study's novel paradigm. Enabling end-to-end security and privacy of cloud-IoT user data and servers in a multi-cloud context is the main goal of this project. This paper discusses the three essential phases of this approach, which improves data privacy and operational efficiency. We start by examining new cloud-IoT users via a three-factor authentication procedure. These users are registered with Trusted Authenticated Servers (TAS), authenticated with their password and user ID, confirmed with photo tags and concealed lines, and have their biometrics examined. Verified users get encryption/decryption keys generated by the Improved Key Generation Process (IKGP). Second, the Trading based Evolutionary Game Theory (TEGT) method is used by the TAS to choose the best DEdS in the Secure MLFL Entities Selection process. We used a machine learning system called Improved Artificial Neural Networks (IANN) to choose our clients. Third, MLFL for Homomorphic Data Sharing and Storage with Secure Privacy Awareness. It employs the Homomorphic Encryption Responsive Lightweight Residual Network with Energy Valley Optimizer (HER-LresNet-EVO) for Local Model Generation. In Global Model Aggregation, Lightweight Factorized Pyramidal Networks (LFPN) are employed. Using MDCS, Super Model Aggregation & Secure Distribution makes aggregated models available to cloud-IoT users. These models are securely kept in the cloud database. Metrics like reduced malicious traffic based on an increase in users, enhanced model accuracy across epochs and rounds, faster encrypting data in relation to data size, optimized homomorphic encryption activities within noise budgets, and a decrease in risks to privacy in comparison to attack ranks are used to gauge the success of the study.

Index words: cloud computing (CC), Distributed Edge Server (DEdS), authentication token (AuthTok), Smart Contract Agents (SMAs), Lightweight Factorized Pyramidal Networks (LFPN), Improved Artificial Neural Networks (IANN)

1. Introduction

Cloud Computing was regarded as the distributed and decentralized software technology that provided on-demand services to users based on their demands over the internet. Simply speaking, the CC was based on the "pay as you go" basis. CC opens up an enormous market on information technology which paved the way for many Cloud Service Providers (CSP) to enable resilient services to the underlying users. Development in information technology is CC. As a result, many commercial applications need new technologies to store or handle large amounts of data. An innovative way to store and

operate apps is through the cloud. It gives consumers access to practically infinite processing power and may have benefits in the form of simple accessibility, scalability, and resource sharing. Webmail, social networking sites, online business apps, and online file storage are all part of the cloud services [1,2]. On-demand self-support, wide network connection, resource pooling, rapid adaptation, and controlled application are some of CC's primary features. Customers (individuals or enterprises) may buy and control their machine services via on-demand self-service. The distribution of services via public or private networks is made possible by access to pooled resources. Cluster service customers are often drawn from a range of computer resources spread across many data centers. Rapid adaptation makes it possible to change programs more or less. The Internet of Things (IoT) enables intelligent online communication and exchange between consumer gadgets.

The applications of adopting IoT technologies such as grid systems, health care, precision agriculture, smart home systems, etc., [3,4]. Integrating the CC and IoT termed the Cloud-IoT environment will provide great benefits to future Internet society. To be more specific, the Cloud-IoT paradigm enables users with one dome, resilient, reliable, secure, and timely services. However, future technology needs much intelligence to optimize workloads and emergencies. For that, Artificial Intelligence (AI) came into the picture which utilizes machines to think like humans and optimize our workload [5,6]. Whereas, acquisition and collection of data for training the AI machines was a major concern as it directly relies on privacy and reliability. Consumers are charged by the product use assessment. The use of its technologies by users (companies, people, etc.) is restricted even though cloud storage has developed into an advanced model of service because of worries about the lack of privacy of their data. Numerous benefits of the cloud include dependability, flexibility, unlimited storage, and practicable collaboration [7]. Despite the cloud's many benefits, it doesn't adequately protect the data that is kept there. Due to the lack of resource availability and the possibility of malicious inside attackers stealing data from a single cloud, the practice of storing enormous data in a single cloud is less common [8].

Information like military or medical records has to be highly secure. Also tried to secure the huge data in a multi-cloud architecture to address the challenges mentioned. To ensure data security, many researchers focused on inter-cloud, multi-cloud, or cloud-of-cloud technologies [9]. There cannot be a single service provider that guarantees total security for the data that is being stored. The customer must look for a multi-cloud that offers availability, confidentiality, and integrity to secure critical data. Utilizing services from various cloud service providers is known as a multi-cloud (MC) environment. A single web interface is provided by a multi-cloud environment to access resources from several cloud platforms. The capacity of the MC to facilitate data exchange will be very beneficial to data users. The MC provides cloud data owners with the freedom to share their data. The main advantage of using several clouds to store large amounts of data [10–13]. A privacy-preserving technique called homomorphic encryption (HE) protects user data by applying certain mathematical calculations on the encrypted data. So that, underlying users in the Cloud-IoT environment performing FL operations did not concern about their data privacy and leakage. However, the prior HE methods were highly prone to computation, communication, and noise budget problems [14,15]. So that many research works adopted Machine Learning (ML) algorithms with the HE method to overcome the issues. The ML method with HE was utilized for performing arithmetic operations which reduces the noise budget and limits the operations over a certain limit

whereas the conventional ML-based HEs were faced with computation overhead problems. On the other hand, malicious insiders are also a serious threat in the Cloud-based IoT environment that caused several cyber security issues such as DDoS attacks, Spoofing attacks, etc..,. To address such problems, the proposed work uses lightweight deep learning-based homomorphic encryption methods and multi-level federated learning (MLFL) to allow an end-to-end solution that assures safety and confidentiality to the multi-cloud-based IoT environment.

1.1. Motivations and objectives

Using homomorphic encryption to protect privacy in the cloud computing environment presented significant problems in terms of security concerns as well as computation and communication costs. Even yet, a large portion of the research on cloud security starts with federated learning or homomorphic encryption. Nevertheless, research on full end-to-end security that takes into account all potential hazards has not yet been conducted. The primary issues with the current research projects are listed below,

- **Increased Primary Vulnerabilities & Data Poisoning:** Most of the prior works ensure the legitimacy of the users by utilizing their basic credentials (ID, password, etc..,). In addition to that some of the work lacks in considering the user legitimacy leads to increased primacy vulnerability threats and malicious traffic. Furthermore, all the existing works lacked analysis of the user's raw data which also leads to data poisoning attacks.
- **Enlarged Noise Budget**: The existing works utilized in this research adopted partial and fully homomorphic encryption methodologies without considering the optimal operations on the encryption data leading to an enlarged noise budget and cost constraints. Although some work adopts machine learning-based fully homomorphic encryption for limiting the noise budget but limits with higher computation and communication overhead respectively.
- **Poor FL Performance:** A centralized cloud server was used for global model aggregation in all previous efforts, which results in single point of failure problems. Additionally, model poisoning assaults were not known in the previous research. Even though they perform secure aggregation using homomorphic encryption but lack of monitoring and verifying the FL rounds might affect the performance of the FL.

This study's primary objective is to provide end-to-end security and privacy for servers and cloud-IoT user information in a multi-cloud context. The following are some of the sub-objectives of this study,

- To reduce the primary vulnerability threats and malicious traffics, we perform novel three-factor authentication based on several credentials and challenging tasks.
- To improve MLFL performance, security, and reliability, we perform MLFL entity selection based on game theory and machine learning algorithms. In addition to that, the data poisoning attacks were also resisted by performing data validation.
- To ensure the data privacy of the users, we adopt lightweight DL-based HE applied on the local, global, and supermodels thereby overcoming the issue of privacy leakage.
- To resist model poisoning attacks, we perform secure and lightweight DL-based aggregation which also reduces the computation, and communication cost.

A. Research Contributions

Below are some of this research's main contributions,

- For limiting malicious traffic in the multi-cloud environment, we perform novel three-factor authentication based on the factors such as some users know, something users have, something users are, and something users possess. The adoption of three-factor authentication also overcomes the primary vulnerability threats.
- For enhancing the performance of the MLFL, we perform secure MLFL entity selection in which both the DEdS and cloud-IoT users were selected based on the security and other parameters using TEGT and IANN algorithms respectively.
- For limiting the noise budget and optimizing the communication and computation overhead, we have adopted LResNet-EVO-based HE on the models of the proposed MLFL which directly applies to the arithmetic operations of the HE.
- For mitigating the model poisoning attacks in the MLFL model training, we perform SMA and LPFA-based secure aggregation in which the models which did not follow the smart contracts were pruned out.

B. Research organizations

The remainder of this document is divided into the following sections: The literature review of the prior work that is more relevant to our work is illustrated in Section II. The primary problem statements that are tackled in the existing works are presented in Section III. The research technique for the suggested work is presented in Section IV and includes a protocol, a mathematical representation, and a pseudocode. The experimental findings and a comparison of the suggested and current works are described in Section V. The suggested study is concluded in Section VI, which also provides plans for this research's future work.

2. Literature Survey

This section deals with the survey of literature on enabling end-to-end security and privacy to the cloud-IoT user data and servers in the multi-cloud environment, the membership proof for FL is described by the authors in [16], who use cryptographic accumulators to produce membership proofs by compiling user IDs. Users can verify the proofs because they are published in a public blockchain. Researchers suggest Privacy-Preserving Federated Learning Membership (PFLM), including membership evidence. PFLM keeps the security guarantees while releasing the assumption of the threshold. To confirm the accuracy of results that have been aggregated from the cloud server, they also create a result verification technique that is based on a particular ElGamal encryption version. The authors of this study use a HE technique based on federated learning to protect data privacy. A single server and many clients are among the entities participating in this endeavor. This study uses a machine learning approach called the multi-layer perceptron algorithm to aggregate the global model. The client data, which was produced by a neural network, was regarded as the local model data. For multi-layer perceptron-based global aggregation, the cloud server received the created local model. Lastly, the clients were given access to the Paillier algorithm, which was used as a HE technique [17]. Pallier homomorphic encryption is being used to protect data privacy across cloud computing environments. Cloud servers (i.e., storage and proxy servers, respectively) and client-side apps are among the entities involved with this endeavor. The client-side applications encrypted their data using RSA public key cryptographic algorithm. After that, encrypted data was homomorphically using the pallier encryption method. On the cloud server side, the proxy server performs re-encryption and is securely stored in the storage server [18]. Integrating filtering and partial homomorphic encryption techniques to protect privacy in cloud computing settings. Sensor nodes,

manager nodes, collector mode, and inquiry nodes are some of the entities that are engaged in this activity. The management and collector nodes, respectively, sent the detected data from the sensor component to the inquiry node. The privacy of the communication was ensured by adopting the Kalman filter and Paillier homomorphic encryption respectively based on the estimation methodologies such as query alliance, cloud alliance, and sensor alliance. The results show that the provided privacy estimation methodologies ensure better than the conventional methodologies [19]. In this study, the authors use a machine learning technique to defend against poisoning assaults in a cloud environment that is based upon federated learning. Random Forest was the machine learning algorithm that was used. The parties engaged in this endeavor include rectifiers, key suppliers, centralized cloud servers, and island users. Keys were being obtained from the key supplier by the island consumer, rectifier, and cloud server. The island user encrypts its models using the keys, and the rectifier encrypts their data before sending it to the cloud server. A random forest approach is used by the cloud server to securely aggregate that model, which is then sent to the island users and rectifiers [20].

Authors in [21] propose homomorphic encryption and momentum-based federated learning algorithm for ensuring data privacy in the cyber security environment. The entities that are participating in this endeavor include cloud servers, trusted computer systems, base stations, and local agents. In order to pre-process the gathered data, normalization, cleaning, the rebuilding process, and sliding windows were all carried out, respectively. Eventually, the trusted server generates a key pair for securely training the local model. The encrypted local models were provided to the cloud server in a homomorphically encrypted manner using the CKKS method. And finally, the model is aggregated on a cloud server, where it may be safely updated and sent to clients. Authors in [22] introduce the resisting the privacy issues in the multi-cloud environment using homomorphic encryption methodology. The entities participating in this endeavor include cloud-assisted laboratories and hospital patients. To address key tampering difficulties in a multi-cloud context, this study uses a multi-key homomorphic technique. For the purpose of diseases prediction, the patient's medical records were homomorphically encrypted and sent to the multi-cloud lab. The encrypted results were provided to the patients where they performed decryption of encrypted homomorphic data. The authors in this work adopts a homomorphic encryption method for providing data privacy for the mobile-based Internet of Things environment. Data users, data owners, and cloud servers are some of the entities participating in this activity. The cloud server first creates a key pair for each user and data owner. After that, queries were asked by the data user for the encrypted data of the data owners. Once that query was determined as valid, the cloud server applies a partial homomorphic encryption method and provides it to the queried user. The user then decrypts it user their private key [23].

The authors of [24] describe a new strategy for very efficient FL with robust privacy preservation in CC. They provide a simple encryption method that maintains adequate model usefulness and privacy. Additionally, training efficiency is increased by the employment of an efficient optimization technique. Using the threat model presented, they show that the proposed method is safe for servers that are honest yet inquisitive and for extreme cooperation. The authors of [25] aim to create a model for privacy preservation in a cloud setting utilizing emerging artificial intelligence capabilities. However, by hosting the data, cloud computing offers organizations enormous advantages. The data cleansing and restoration stages are the two main components of the system. The combined meta-heuristic method also determines the best key

generation efficiency of the proposed sanitization technique. Jaya-based Shark Smell Optimization (J-SSO) is a hybrid algorithm that combines two powerful methods, such as Shark Smell Optimization (SSO) and Jaya Algorithm (JA). The authors of [26] provide a novel data preservation strategy based on the Chinese Remainder Theorem (CRT) for protecting user data in cloud databases. For accessing the encrypted data kept in the cloud databases, a brand-new key management system that was developed with CRT is used. Dual encryption methods with innovative decryption algorithms and innovative first and second encryption approaches are used in the suggested CRT-based safe storage solution. The authors in [27] introduce a Novel Data Privacy-Preserving Protocol (NDPPP) for Multidata Users using algorithms based on evolution. The data owner encrypts the files before sending them to the cloud. The cloud service provider enables users to download safe files efficiently and without losing any data. The authors in [28] suggested a lightweight homomorphic encryption-based privacy-preserving technique for the Internet of Things. They investigated and examined privacy issues between data consumers, data owners, and dubious third-party cloud services. Meanwhile, to protect the privacy of data users, very computationally efficient homomorphic algorithms are proposed.

I. PROBLEM STATEMENT

The special problems that contemporary works often face are the main topic of this section. Additionally, the recommended cure is given. Several of the specific problem statements that already exist include,

Background of existing problems: Authors in [29] adopt FL and homomorphic encryption methodologies for ensuring the privacy of the cloud environment. This study uses a privacy-preserving federated learning method, which depends on a homomorphic algorithm, to achieve this. Additionally, this study uses the distributed homomorphic encryption approach, which partly encrypts and decrypts data. Participants in the initiative include cloud users, centralized cloud servers, compute servers, and key provisioning authorities. The cloud user was registered and authenticated to the key provisioning authority and acquired private and public keys. The generated keys were then provided partially to the cloud server and server for computation. With that partial key pairs, the cloud server and server for computation perform joint training of the global model and are securely shared with the end users. The problems employed from these approaches are,

- This work performs authentication of cloud users based on the basic credentials. However, it was easy for the high-potential attackers to crack the authentication procedure and performed malicious insider operations.
- The centralized cloud server was responsible for global model aggregation based on distributed homomorphic encryption method. However, the lack of considering the model behavior and deviations leads to model poisoning attacks, and the adoption of centralized servers also leads to single-point-of-failure issues.
- Furthermore, the secure federated learning was taken place without performing optimal client selection which affects the accuracy of the federated learning model as it cannot withstand many clients on a single round.

To secure the cloud environment, the authors of [30] conduct secure access policies based on homomorphic encryption. This work's homomorphic encryption approach is known as the secure partially homomorphic encryption algorithm. Cloud users, key generation entities, and elastic beanstalk are among the entities involved in this effort.

Key pair generation, encryption, decryption, and homomorphic operations were all handled by the secure partially homomorphic encryption algorithm. Based on it, the cloud-based program includes policies to limit unauthorized users, such as a decryption policy, an access policy, an assessment policy, and an administration policy. The proposed encryption methodology outperforms the current methods. The issue employed in this work are,

- In this case, cloud customers could only apply four different kinds of access control rules to protect the privacy of their data. To restrict the amount of harmful traffic in the AWS-based cloud environment, further work must be done.
- In the AWS cloud environment, data and communication privacy was ensured by the use of the partial homomorphic encryption approach. However, there are tampering difficulties since partial homomorphic encryption only permits a restricted amount of operations against encrypted data.
- In addition to that, encryption and homomorphic encryption were applied to user data without performing analysis of the data which leads to data poisoning attacks and paved the way for illegal activities.

Authors in this work adopt dual layer homomorphic encryption methodology for ensuring privacy in cloud computing environments. Cloud users, administrators, cloud storage, and cloud servers are among the entities participating in this endeavor. The procedures used in this project, including the encryption and key creation processes, respectivel. Both the processes composed of dual layers of homomorphic encryption were applied to ensure privacy including a lightweight cryptography-based encryption layer and multiplicative homomorphic method using Rivest Shamir Adleman algorithm [31]. Some of the major problems employed in this work are,

- Even though this work performs key generation for encrypting the cloud users' data. However, the lack of considering the user's legitimacy and authenticity leads to increased malicious traffic and unauthorized access.
- Furthermore, this study uses the Rivest Shamir Adleman method to accomplish multiplicative homomorphic encryption. However, since the Rivest Shamir Adleman method needs both symmetric and asymmetric keys, it was frequently broken.
- Furthermore, the security of the server which shares and stores the data and services are left unfocused in this work leading to increased privacy threats to user data. More specifically privacy related attacks were easily taken place such as hijacking and poisoning attacks respectively.

The authors of [32] use a homomorphic encryption approach based on active and federated learning to ensure data privacy in a client-server setting. The entities participating in this endeavor include many clients and a server. The clients were given a public key to encrypt their local model. Individual local models were trained using an active learning approach in which the unlabeled instance was predicted from the labeled data during the active learning query phase. The server used a weighted aggregation strategy and a homomorphic encryption technique based on Brakerski Fan Vercauteren to safely deliver the client's data while maintaining their privacy after receiving the trained encrypted local models for global model aggregation. The issues with this strategy include,

- Here, the single server multiple client scenario was considered whereas considering that single server scenario leads to increased single point of failure thereby leading to data loss and data unavailability issues.
- This work utilizes Brakerski Fan Vercauteren based homomorphic encryption method for ensuring the privacy of the client data. However, adopting Brakerski Fan

Vercauteren based homomorphic encryption lacks with providing security against adaptive cyber security attacks.

Research solutions: The study provides a fresh approach to the problems that CC is now facing. The proposed study develops a strong three-factor authentication system that improves security by merging numerous credentials such as user ID, password, IP, location, image tag, and biometrics. This complete technique efficiently thwarts insider attacks. Furthermore, the research addresses model poisoning concerns by implementing smart contract agent-based secure aggregation. This technique protects the integrity of aggregated models by leveraging indicators such as model consistency and misbehavior. Furthermore, by optimizing entity selection, the research strategically improves the MLFL process. Clients' DEdS are carefully selected based on many variables using the TEGT and IANN algorithms, respectively. To combat fraudulent traffic, the study used a comprehensive three-factor authentication method that includes factors the user knows, has, is, and owns. This all-encompassing authentication technique efficiently prevents illegal access and threats. The proposed study also introduces HER-LResNet-EVO, a unique homomorphic encryption mechanism. This solution addresses the difficulties of a restricted noise budget and computing overhead during arithmetic operations. It also uses the IANN method to assure data uniqueness during client selection. The research strengthens security against model poisoning attacks in the MLFL process by using SMA and LFPA. The suggested approach efficiently manages local model and global model aggregation by utilizing a distributed edge server and multi-cloud environment, avoiding issues related to single points of failure, data loss, and unavailability. Lastly, the research offers a thorough framework that greatly enhances the effectiveness and security of the cloud computing environment by optimizing entity selection and aggregation procedures in addition to bolstering security via complex authentication and encryption mechanisms.

II. PROPOSED METHOD

Using AI algorithms for federated learning and homomorphic encryption, respectively, to ensure privacy in cloud computing environments is the primary focus of this study. The proposed work's general design is shown in Fig. 1. The proposed work adopts a deep learning-based fully homomorphic encryption methodology for reducing the existing issues such as computation, communication overheads, and limited/unlimited noise budget. Furthermore, we have also adopted Multi-Level Federated Learning (MLFL) for maintenance, fault tolerance, and effective data sharing in the proposed cloud environment. The major entities involved in this work such as Trusted Authenticated Servers (TAS), Cloud-IoT users, Distributed Edge Servers (DEdS), Smart Contract Agents (SMAs), Multiple Decentralized Cloud Servers (MDCS), and Super Aggregator (SA). One can easily grab the research core idea by skimming and scanning the below-mentioned sequential research processes,

- Three-Factor Novel Authentication
- Secure MLFL Entities Selection
- MLFL-based Secure Privacy-Aware Homomorphic Data Sharing & Storage

A. Three-Factor Novel Authentication

Initially, the cloud-IoT users and DEdS need to ensure their legitimacy and authenticity to reduce the primary vulnerability threats and unwanted malicious traffic. For that, the proposed work adopts a three-factor novel authentication solution in which only the cloud-IoT users are verified and validated through three-factor authentication whereas the DEdS are authenticated based on the conventional methods. The cloud-IoT

users are initially registered to the TAS with their user ID, location, IP address, biometrics (i.e. finger veins and eye veins), and picture tag (i.e. cloud-IoT users are advised to provide any images which are not taken from any open source platform.

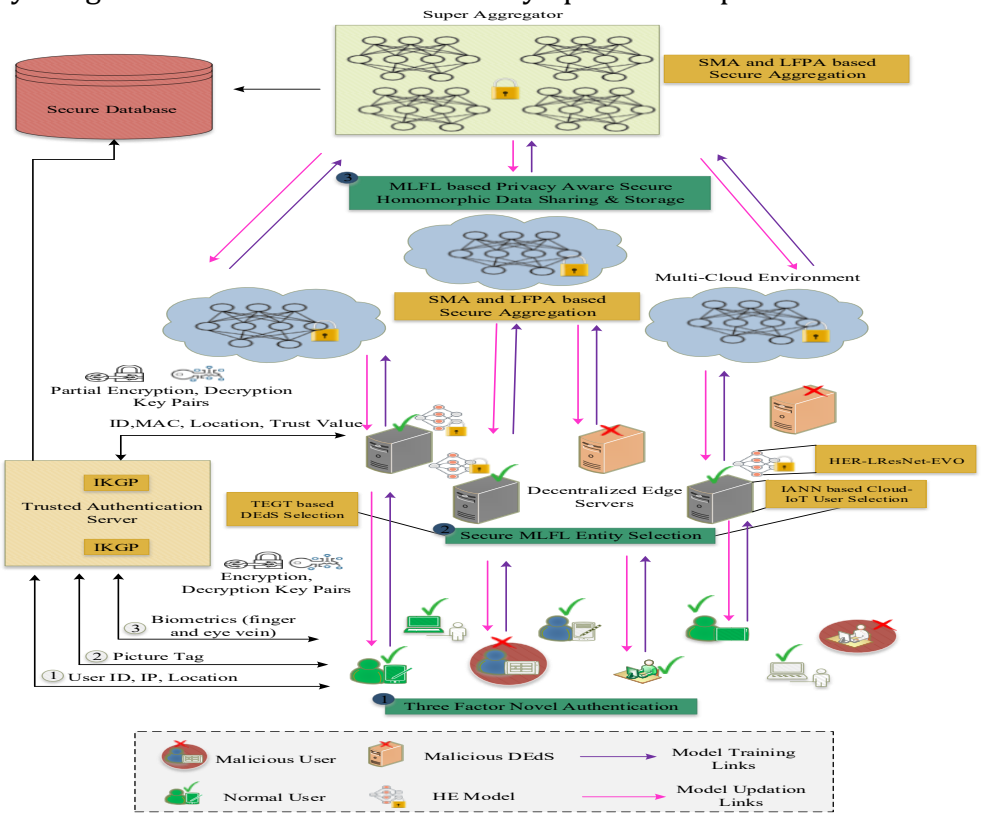

Fig. 1 Overall architecture of proposed work

Along with that image, the users are advised to write a secret two-word secret sentence). Once the registration is completed, the TAS provides the Authentication Token (AuthTok) as the symbol of successful registration. The AuthTok is utilized for initiating the authentication. During authentication, the cloud-IoT users need to pass the three levels of the authentication stage. In the first stage, the TAS asked the user to verify their user ID and password. In the second stage, the TAS provides a challenge to the users to select the picture tag and provide the appropriate secret sentence that the user provided at the time of registration. In the third stage, the TAS asks the user to submit any one of the biometrics. Once the biometrics is verified as correct, an OTP is generated and the users are advised to enter the OTP to the TAS for successful authentication. For successfully authenticated users, the TAS generated a public and private key pair using **Improved Key Generation Protocol (IKGP)** for encryption and decryption respectively.

A single key is utilized by the symmetric block cipher IKGP for both encryption and decryption. Sequences of 128 bits are used as both the IKGP's input and output. This method uses a key that is made up of 128, 192, or 256 bits. 8-bit bytes are used by IKGP. Applying the polynomial representation shown below, these bytes can be described as the components of a finite field:

$$f(x)= \beta_{n-1}\chi^{n-1}+\beta_{n-2}\chi^{n-2}+\ldots+\beta_1\,\chi+\beta_0=\sum_{i=0}^{n-1}\beta_i\,\chi^i \qquad (1)$$

where every β_i has a value of either 0 or 1.

The 4×4 state matrix contains the 128-bit IKGP input block. The variable β_{ij} represents the matrix's elements, with $0\le i, j\le 3$, and i, j denoting the column and row numbers,

respectively. IKGP allows for various rounds depending on the size of the bits in the keys. We decided on a 256-bit key size notion for our inquiry; hence, we needed 14 rounds, represented by the letter $\Im$. Each round's keys are also provided using the IKGP key scheduling mechanism. Because of the way the key sequencing system was built, it is possible to identify the initial input key that generates the round keys by exposing any round key. Multiple round transformations are applied to the input state matrix. As it passes through many cipher phases and ultimately produces the ciphertext, the state matrix changes. The steps in an IKGP round are as follows.

Sub bytes: The IGKP includes a nonlinear step like this. It applies an S-box on the state matrix's bytes. The multiplicative inverse of each byte in the state matrices is substituted for it, then an affine mapping is applied:

$$\beta_i' = \beta_i \oplus \beta_{i+4\ mod\ 8} \oplus \beta_{i+5\ mod\ 8} \oplus \beta_{i+6\ mod\ 8} \oplus \beta_{i+7\ mod\ 8} \varsigma_i,\ for\ 0 \leq i < 8 \quad (2)$$

where ς_i is the ith bit of a byte with the value 63 and β_i is the ith bit of the byte. As a result, the relationship between the input byte χ and the output byte $\mathbb{Y}$ of the S-box is $\mathbb{Y} = \alpha.\chi^{-1} + \gamma$, where α and γ are constant matrices.

Shift rows: A state matrix shifts its last three rows by a certain number of bytes. Here's the process that's used:

$$\delta'_{\gamma,\varsigma} = \delta_{(\gamma(\varsigma + shift(\gamma+\varphi) mod \varphi)} \quad (3)$$

For $0 < \gamma < 4$ *and* $0 < \varsigma < \varphi$

where φ is the state matrix's word count. Assuming a 128-bit input size and a 4×4 state matrix, AES always uses $\varphi = 4$. The state matrix's cells are identified by the letters δ, row γ, and column ς, respectively.

Mix columns: This modification works column-by-column on the state matrix, assuming each column as a four-term polynomial over ϱ (2^8) multiplied $\chi^{4+1}$by a fixed polynomial $\alpha(\chi)$, provided by

$$\alpha(\chi) = \{03\}\chi^3 + \{01\}\chi^2 + \{01\}\chi^1 + \{02\} \quad (4)$$

The method for multiplying state matrix columns is provided by

$$\delta'(\chi) = \alpha(\chi) \otimes \delta(\chi) \quad (5)$$

where $\delta(\chi)$ is a state in the state matrix and $\otimes$ is polynomial multiplication modulo.

Add Round Key: A straightforward bitwise XOR operation is used in this procedure to add a round key to the state. The size of each round key is determined by φ from the key schedule. Each of those φ is added to a column in the state matrix to fulfill the following requirement:

$$[\delta'_{0,\varsigma}, \delta'_{1,\varsigma}, \delta'_{2,\varsigma}, \delta'_{3,\varsigma}] = [\delta_{0,\varsigma}, \delta_{1,\varsigma}, \delta_{2,\varsigma}, \delta_{3,\varsigma}] \oplus [\omega_{round \times \varphi + \varsigma}] \quad (6)$$

Where $0 \leq Round < \Im$ round is the round value at which the round key is inserted, and $\oplus$ are bitwise XOR.

Except for the final round, all these stages are carried out for each AES round. The last round skips the MixColumn step. The key expansion procedure creates round keys, and these are one of the most crucial elements of the round function stages. An initial set φ of is required for the procedure, and of crucial data are required for any of the $\Im$ rounds. The sum obtained by the key expansion is ($\Im$ + 1). A linear array of 4-byte words makes up the generated key schedule, which is represented by the numbers ω_i, $0 \leq i \leq \varphi$ and ($\Im$

+ 1). These 4-byte words are input into the function SubWord (), which then applies S-box to each word. Circular permutations are carried out using another function called Rotword (). In the equation below, the values written [χ^{i-1}, {00}, {00}, {00}] using χ^{i-1} powers of χ are retained in the round constant array Rcon[i].

$$Rcon[\mathrm{i}]= \chi^{\mathrm{i}-4/4} \bmod (\chi^8+\chi^4+\chi^3+\chi+1), \tag{7}$$

The key expansion process for 256-bit keys ($\Re$=8) varies separately from the processes for 128- and 192-bit keys, wherein i is the current round. SubWord is applied to ω [i-1] before the XOR if i-4 is an integer of $\Re$ and $\Re$ is equivalent to 8. $\Re$ stands for the quantity of 32-bit words in a key. Eventually, the DEdS also registered and authenticated to the TAS based on their ID, MAC, location, and trust value. The authenticity of the DEdS is ensured based on their credentials and trust value variations. Similarly, the TAS provides partial encryption and decryption key pairs to the DEdS using IKGP for performing data filtration on user data.

B. Secure MLFL Entities Selection

The entities that will take part in the MLFL are carefully chosen to improve the MLFL's general efficiency after authentication. For that, the proposed work performs DEdS selection and clients. The reason for selecting both entities is to ensure security and reliability during MLFL rounds. To be more specific, the DEdS is responsible for local model generation henceforth selecting the appropriate DEdS must be ensured to mitigate the model poisoning attacks. The TAS uses a **TEGT** algorithm to choose the best DEdS based on parameters including energy level, capacity, trustiness level (i.e., both direct and indirect trust), and previous behaviors. TEDT is frequently utilized to research how various tactics or behaviors might develop and endure in a population as a result of natural selection. We are using evolutionary game theory to analyze the interactions between DEdS agents and TAS agents.

Strategies and Payoff: Each agent (TAS and DEdS) in the setting of the challenge has a set of strategies that determine how they will respond in the interaction. These tactics can be visualized as an array of probabilities that an agent will take a specific action. The value or benefit an agent derives from the selected approach is known as the payout. The payoff υ_{ij} that agents receive i when playing strategies $\mathbb{S}_j$ influences by the interaction between them. It can be the combination of different factors energy level ($\mathcal{E}$), capacity ($\mathcal{C}$), $Trustiness$ ($\mathcal{T}$):

$$\upsilon_{ij} = f(\mathbb{S}_i, \mathbb{S}_j\, \mathcal{E}, \mathcal{C}, \mathcal{T}) \tag{8}$$

Replicator dynamics: The replicator dynamics equation explains how the population's strategy frequencies evolve. The frequency's rate of change $\mathcal{X}_i$ of strategy $\mathbb{S}_i$ in the population is proportional to the difference between the average payoff of strategy $\mathbb{S}_i(\upsilon_i)$ and the average payoff of all strategies ($\hat{\upsilon}$):

$$\frac{d\mathcal{X}_i}{dt}=\mathcal{X}_i.\,(\upsilon_i - \hat{\upsilon}) \tag{9}$$

Fitness calculation: The fitness $\mathcal{F}_i$ of a strategy $\mathbb{S}_i$ is determined by the cumulative payoff the agents receive when playing against all other strategies in the population:

$$\mathcal{F}_i = \sum_{j\neq i} \upsilon_{ij} \tag{10}$$

Selection probabilities: The probabilities ρ_i The likelihood that an agent will be selected for reproduction depends on its fitness $\mathcal{F}_i$ about the population's overall fitness:

$$\rho_i = \frac{\mathcal{F}_i}{\sum_k \mathcal{F}_k} \tag{11}$$

By using this equation (11), the idea of natural selection is mimicked in that agents with higher fitness values are more likely to be chosen for reproduction.
Strategy update: The population's agents adjust their tactics to reflect the new ones produced by reproduction and mutation. This stage displays how agents' actions alter in response to others' shifting tactics.

$$\mathbb{S}'_i = \mathbb{S}_i + \mathcal{M} \tag{12}$$

where " $\mathcal{M}$ " refers to the alterations brought about by the mutation process. This equation describes how the mutation process alters the strategies of individual agents. Frequency update: The frequency of strategy $\mathbb{S}_i$ in the population at the next generation ($\mathcal{X}'_i$) is updated based on the replicator dynamic equation. The change in frequency $d\mathcal{X}_i$ is proportional to the difference between the average payoff of strategy $\mathbb{S}_i$ and the average payoff of all strategy $\hat{v}$:

$$d\mathcal{X}_i = \mathcal{X}_i . (v_i - \hat{v}) \tag{13}$$

Equilibrium conditions: In an evolutionary game, an equilibrium is reached when the population's frequency of different strategies stays constant across time. Setting the rate of change equation to zero will reveal this:

$$\frac{d\mathcal{X}_i}{dt} = 0 \tag{14}$$

Pseudocode for TEDT
Initialize strategies **and** payoff
While not equilibrium:
  Calculate v_{ij} **for** $\mathbb{S}_i$
  Evaluate $\mathcal{F}_i$ **for** $\mathbb{S}_i$
  Calculate ρ_i
  Reproduction **and** mutation
  for each agent
   Choose parents based on ρ_i using eq. (11)
   Apply crossover and mutation based on $\mathcal{X}'_i$ using eq. (12)
  Update $\mathcal{X}_i$ **for** each $\mathbb{S}_i$
  Check **for** equilibrium using eq. (14)
Output optimal DEds selection

When a population reaches equilibrium, the average payoff for all methods equalizes and the population achieves stability. The TEGT is performed among TAS and DEdS in which TAS acts as a seller and DEdS acts as a buyer. The selected optimal DEdS are responsible for selecting the optimal clients. For client selection, the proposed work adopts a machine learning algorithm named **Improved Artificial Neural Networks (IANN)** based on the metrics such as energy level, bandwidth, trust, CSI, RSSI, communication efficiency, and data originality. Only optimal and secure clients are involved in the MLFL process.

There are three layers in the IANN architecture: input, hidden, and output. Every neuron in the input layer has a corresponding criterion used for client assessment. Each client's selection score is produced by the output layer, and intricate relationships between metrics are captured by the hidden layers.

Input layer: The input layer of the IANN is made up of neurons that reflect the many parameters used to assess consumers. Every metric is a numerical value that represents a client's characteristics. The following is how the input metrics will be displayed: energy level, bandwidth, data uniqueness, CSI, RSSI, trust, and communication effectiveness.

Hidden layer: The neural network's hidden layers record intricate connections between the input data. Activation functions are used in these layers to provide nonlinearity despite transforming the incoming data linearly. Let's call the weighted sum at the i^{th} hidden layer's j^{th} neuron y_{ij}:

$$y_{ij}=\sum(\omega \times \mu)+ \phi \quad (15)$$

ω refers to the weights that connect the neurons in the current layer to those in the preceding layer. μ represents the neurons' outputs from the previous layer. ϕ indicates the present neuron's bias term.
After using an activation function (ϱ), the j^{th} neuron in the i^{th} hidden layer produces the following output:

$$\Phi_{ij}= \varrho(y_{ij}) \quad (16)$$

Output layer: To determine whether a client is eligible to take part in the MLFL, the output layer produces a single output score. To get this score, the outputs of the last hidden layer are sent via the output layer's neurons.

$$Output\ score = \sum(\omega \times \Omega)+ \phi \quad (17)$$

Ω refers to the outputs from the last hidden layers.

In order to provide input measurements and generate an output score that establishes MLFL eligibility, the IANN architecture makes use of hidden layers, activation functions, and linear transformations. The neural network accumulates its weights and biases during training to enhance selection. Metrics are present in the input layer. progressively, input measurements pass through the concealed levels. Weights, biases, and activation functions are used by each hidden layer neuron to process information. The network transforms processed input across hidden layers to collect relationships. The output layer receives the processed data at the end and produces the output score. A number of weighted computations and activation methods are needed to transmit input measurements across hidden layers and produce an output score. Based on provided measurements, the forward transmission aids the network in determining the suitability of the client. Fig. 2 demonstrates how the neural network's weights and biases are trained using historical data, enabling it to use training data patterns to make tenable client selection judgments. Only the best and safest clientele participate in MLFL.

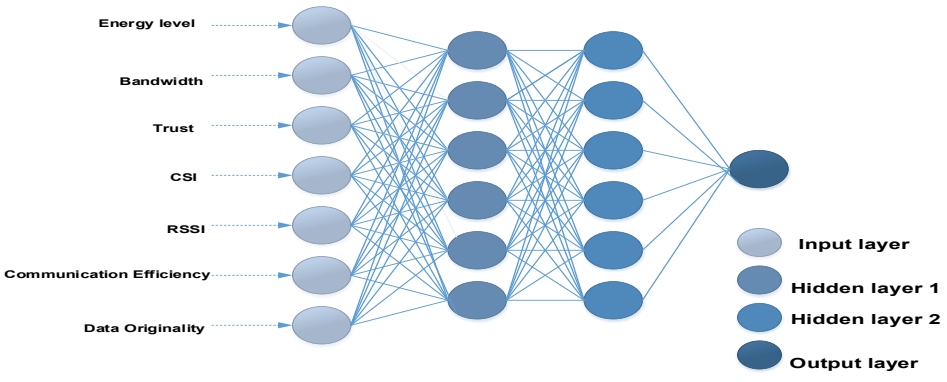

Fig. 2 Architecture of IANN

C. MLFL-based Secure Privacy-Aware Homomorphic Data Sharing & Storage

Once the MLFL entity selection was completed, the process of MLFL gets started in the proposed cloud environment. Since, we adopt MLFL the sub-processes involved in the MLFL such as local model, global model, and supermodel generation & aggregation respectively.

a. Local Model Generation

The DEdS are in charge of creating local models for cloud-IoT users. The chosen DEdS receives the encrypted data from the users in order to train the local model. To protect data privacy without decrypting it, the DEdS encrypts the encrypted data using homomorphic encryption. To be more specific, the proposed work applies a novel optimized deep learning-based homomorphic encryption method named HER-LResNet-EVO. The method we specifically developed to ensure data privacy using the HER scheme is described in this section. With the scaled power activation (SPA) function, reliable optimization is required, to nonlinearly map hidden features in the deep model, we apply a distinctive constrained activation method (i.e., LResNet) and create a flexible HER-appropriate activation function. By merging the popular CNN and dense layer, we developed LResNet, and we showed that it can be quickly implemented into a HER scheme to provide security for a range of models and applications. Fig. 3 demonstrates the overall structure of LResNet. We suggested a partial activation layer to translate hidden attributes to a nonlinear space after the CNN layer. The residue activation layer, a HER-friendly technique, is used to apply the SPA function upon the residue portion of the hidden tensor. Note that the residue activation layer was constructed only using additions and multiplications. This perspective suggests that LResNet and HER may be utilized directly in tandem.

The main function of the activation layer in a deep approach is to identify complicated variables' non-linearity. For difficult jobs, non-linear data may be separated using non-linear models. For this kind of job, neural networks may use the Rectified Linear Unit (ReLU) with Sigmoid functions for non-linear activation, for instance. Such effectively generated non-linear activation functions, however, do not quite meet the need for a HER scheme in previous work, where only adds and multiplication operations have been homomorphically permitted. An activation function, which is used to nonlinearly map hidden layers in deep models, is represented as (q), where ϖ(q) indicates the activation function's input. The SPA function is expressed formally as

$$\varpi(\mathrm{q})=\varkappa.\,\mathrm{q}^{k} \tag{18}$$

where s is the scalar term ($\varkappa$) and the power term (k) enables hidden characteristics to be nonlinear. The SPA function complies with the distinct and continuous activation qualities. To make hidden characteristics zero-centered and properly optimizable, we design a scaled term. It should be noted that the SPA is not always monotonic, which complicates instruction. Another possible problem is the infinite derivative of the SPA function. The derivative of Eq. (18) is easily shown to be $\varpi(\mathrm{q})=\varkappa.\,\mathrm{q}^{k-1}$. We developed a partial activation technique to solve the SPA function's optimization problem, which is explained in the next section.

Effective optimization of a deep network with an SPA function is unfeasible due to the unbound derivative of SPA, even though doing so guarantees that an efficient deep model could be created using non-linear activation functions. Motivated by the

truncation approach of the one-shot learning generative network and the residue network. We provide the following in response to a partially activation strategy (i.e., LResNet) for deep networks. We only use the hidden characteristics that are higher than the global average value, to be more precise. Divergence caused by unbounded derivatives may be readily removed as the global average value stays constant. The letters ı for a layer and q for its input tensor may be used to mathematically describe the partially activated layer.

$$\iota(q) = \varpi(\text{q-}\bar{\text{q}})+ \bar{\text{q}} \tag{19}$$

where z is the average value of the hidden tensors and (q) is the SPA function. Hidden characteristics $\bar{\text{q}}$ in Eq. (19) have been divided into the residue component (z z) and the global average part (q-$\bar{\text{q}}$). Only the residue portion is impacted by the unbounded derivative issue since the SPA function only introduces nonlinearity within the residue portion. Furthermore, the gradient is modestly modified in the corresponding hidden layers. In this sense, the partial activation layer-equipped model is more robust and stable. Using a residue activation layer and a scaled power-activated function, we developed LResNet. We particularly combined the CNN layer and the thick layer with the partial activation layer since they have all been shown to be HER-friendly and are often used in HER evaluation.

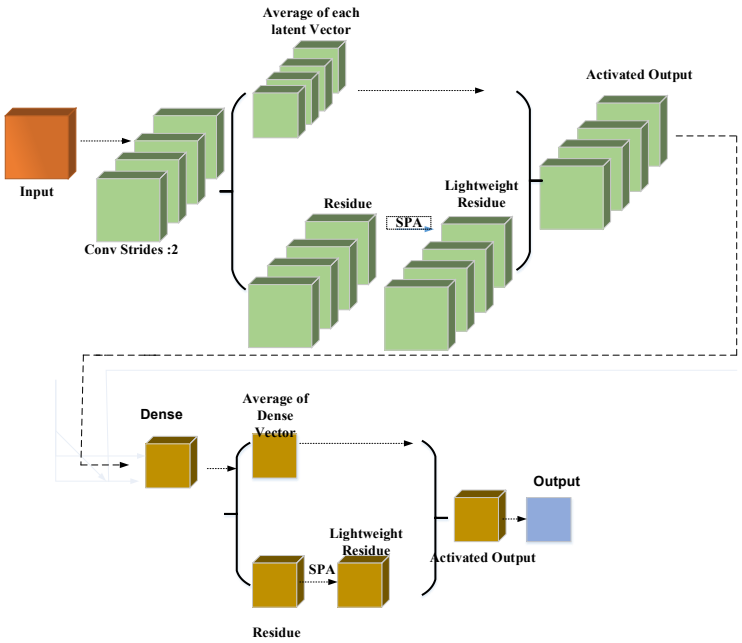

Fig.3 Structure of LResNet

This section provides a detailed description of the EVO as an optimization technique based on physics concepts. In the first step of the initialization process, the solution possibilities ($\mathbb{X}_i$) are assumed to be particles with different levels of stability in the search space, which is assumed to represent a specific area of the universe.

$$\mathbb{X}=\begin{bmatrix}\mathbb{X}_1\\ \mathbb{X}_2\\ \vdots\\ \mathbb{X}_i\\ \vdots\\ \mathbb{X}_n\end{bmatrix}=\begin{bmatrix}\mathcal{X}_1^1 & \mathcal{X}_1^2 \dots \mathcal{X}_1^j \dots \mathcal{X}_1^q\\ \mathcal{X}_2^1 & \mathcal{X}_2^2 \dots \mathcal{X}_2^j \dots \mathcal{X}_1^q\\ & \vdots\\ \mathcal{X}_i^1 & \mathcal{X}_1^2 \dots \mathcal{X}_i^j \dots \mathcal{X}_i^q\\ \mathcal{X}_n^1 & \mathcal{X}_n^2 \dots \mathcal{X}_n^j \dots \mathcal{X}_n^q\end{bmatrix} \tag{20}$$

$$\mathcal{X}_i^j = \mathcal{X}_{i,min}^j + \text{rand.}\ (\mathcal{X}_{i,max}^j - \mathcal{X}_{i,min}^j) \quad (21)$$

The method's second step determines the particles' Enrichment Bound ($\mathfrak{M}$), and this is used to compare the properties of neutron-rich and neutron-poor particles. For this, an objective function assessment of each particle is performed in order to determine its Neutron Enrichment Level ($\mathfrak{P}$). The following is a mathematical presentation of these characteristics:

$$\mathfrak{M} = \frac{\sum_{i=1}^{n} \mathfrak{P}_i}{n}, \text{ i=1, 2,...n.} \quad (22)$$

By evaluating the goal function, the third step determines the particle stability levels:

$$\mathfrak{C}_i = \frac{\mathfrak{P}_i - \mathcal{S}}{\mathcal{W} - \mathcal{S}}, \text{ 1, 2...n.} \quad (23)$$

The particle's best stability levels (S) and (W) include the greatest and worst stability levels across the universe, which correspond to the lowest and highest values of the function objective values defined so far. where $\mathfrak{C}_i$ is the stability level of the ith particle. In order to put the decay process employing alpha, beta, or gamma schemes into viewpoints, the neutron enrichment level of a particle is taken into consideration in the main search loop of the EVO if it surpasses the enrichment limit ($\mathfrak{P}_i > \mathfrak{M}$).

This is done by creating a random integer in the interval [0, 1] that mimics the universe's Stability Bound (ξ). The alpha and gamma drops are thought to occur if a particle's stability value hits the stability limit ($\mathfrak{C}_i > \xi$), as they occur more often for heavier particles with greater stability levels. The best stable level ($\mathbb{X}_{\mathcal{S}}$) of the rays in the particle or prospect replaces the emitted rays, which stand in for the decision criteria in the final candidate.

These components have the following mathematical expression:

$$\mathbb{X}_i^{new\ 1} = \mathbb{X}_i(\mathbb{X}_{\mathcal{S}}(\mathcal{X}_i^j)) \quad (24)$$

A novel solution candidate for the EVO position-updating process is produced by gamma decay, which releases γ rays to improve particle stability. The solution candidate is determined by particle photons. A nearby particle or candidate ($\mathbb{X}_{\mathcal{G}}$) is used to replace excited particles in order to replicate the interaction between them and other molecules or magnetic fields.

The overall range between the particle and others is computed using the nearest particle.

$$\mathbb{Q}_i^k = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (25)$$

The particle characteristics in the search space are (x_1, y_1 and x_2, y_2), and the distance between the ith particle and its kth adjacent particle is $\mathbb{Q}_i^k$. The second potential candidate in this step was created using the following position update procedure:

$$\mathbb{X}_i^{new\ 2} = \mathbb{X}_i(\mathbb{X}_{\mathcal{G}}(\mathcal{X}_i^j)) \quad (26)$$

The position update procedure involves a regulated motion between the candidates having the highest stability level ($\mathbb{X}_{\mathcal{S}}$) and the particle center ($\mathbb{X}_c$). The technique mimics the tendency of particles to move toward the stability band, which is home to well-liked particles and has greater stability levels. These components have mathematical expressions.

$$\mathbb{X}_c = \frac{\sum_{i=1}^{n} \mathbb{X}_i}{n} \quad (27)$$

$$\mathbb{X}_i^{new\ 1} = \mathbb{X}_i + \frac{r_1 \times \mathbb{X}_{\mathcal{S}} - r_2 \times \mathbb{X}_c}{\mathfrak{C}_i} \quad (28)$$

A beta decay-based location update procedure is used to shift particles toward the most stable candidate ($\mathbb{X}_{\mathcal{S}}$) and a nearby contender ($\mathbb{X}_{\mathcal{G}}$) while preserving particle stability in order to improve algorithm exploitation and exploration. These elements are expressed mathematically as follows:

$$\mathbb{X}_i^{new\,2} = \mathbb{X}_i + (r_3 \times \mathbb{X}_{\mathcal{S}} - r_4 \times \mathbb{X}_{\mathcal{G}}) \tag{29}$$

In general, a particle with a lower $\aleph/\wp$ ratio and a lower neutron enrichment level ($\mathfrak{P}_i \sim \mathfrak{M}$) migrates toward the stabilization band via positron emission or electron capture. The table that follows establishes the random movement of the search space to account for these movements:

$$\mathbb{X}_i^{new} = \mathbb{X}_i + r \tag{30}$$

In the search space, $\mathbb{X}_i^{new}$ and $\mathbb{X}_i$ represent the incoming and current location vectors of the ith particle,

although particle mobility is controlled by r, a random value between [0, 1]. Below is a description of the EVO pseudocode:

Pseudocode for EVO

initial position of $\mathbb{X}_i$ as search space particles
Fitness value for initial solution candidates is $\mathfrak{P}_i$
While iterations< Maximum number of iterations
  Evaluate $\mathfrak{M}$ of the particles
  Choose the particle with the $\mathbb{X}_{\mathcal{S}}$
    for i=1:n
      Determine $\mathfrak{C}_i$ of the ith particle
      Assess $\mathfrak{P}_i$ of the ith particle
      if $\mathfrak{P}_i > \mathfrak{M}$
        Determine ξ of the particles
        if $\mathfrak{C}_i > \xi$
          Generate Alpha 1 and 2
          for j=1: Alpha index 2
            $\mathbb{X}_i^{new\,1} = \mathbb{X}_i(\mathbb{X}_{\mathcal{S}}(\mathcal{X}_i^j))$
          end
          Generate Gamma index I and II
          Choose a $\mathbb{X}_{\mathcal{G}}$
          for j=1: Gamma index 2
            $\mathbb{X}_i^{new\,2} = \mathbb{X}_i(\mathbb{X}_{\mathcal{G}}(\mathcal{X}_i^j))$
          end
        else if $\mathfrak{C}_i \leq \xi$
          Evaluate $\mathbb{X}_c$ using eq. (28)
          Determine $\mathbb{X}_{\mathcal{G}}$ using eq. (29)
        end
      else if $\mathfrak{P}_i \leq \mathfrak{M}$
          $\mathbb{X}_i^{new} = \mathbb{X}_i + r$
      end
    end
  end
end while
  Return the particle $\mathbb{X}_{\mathcal{S}}$

end procedure

By directly applying LResNet-EVO to the HE's arithmetic operations, the computation complexity and communication overhead are decreased. In addition, EVO is used to choose the right noise budget for the encrypted data. For the purpose of generating and aggregating global models, the homomorphically encrypted data are regarded as the local model.

b. Global model Aggregation

The global model aggregation is carried out in the multi-cloud environment shortly after the completion of the local model development. Global model aggregation is the responsibility of the several cloud servers. The SMAs in the MDCS create a policy based on the framework's consistency and misbehaviors prior to aggregating the local model. To be clearer, during aggregation, the local model must be consistent (i.e., abide by the law of the MLFL aggregation algorithm). Any deviations from the MLFL aggregations are considered malicious and punished. On the other hand, the misbehaviors are considered as intentionally interfering with the MLFL communication rounds, disturbing the communication of other user models. Any cloud-IoT user model that violates the SMAs rules are considered a malicious user and kicked out thereby mitigating the model poisoning attacks. Other than the malicious cloud IoT, the normal models are aggregated using lightweight deep learning named **Lightweight Factorized Pyramidal Networks (LFPN)**.

Layer 1 started in models 1, 2, n, n+1, n+2m, and global model aggregation occurs in the multi-cloud environment. The numerous cloud servers aggregate global models. SMAs in the MDCS generate a policy based on model consistency and misbehaviors before aggregating local models. Use the LFPN module for context integration and misbehavior detection when aggregating local models worldwide. Dilated convolutions can improve aggregation by considering contextual information from several scales. This can increase global model accuracy and robustness by collecting local and global context in the aggregated model. The SA securely aggregates MDCS models to facilitate multi-cloud data sharing, predictive maintenance, and data sharing. Supermodel aggregation follows the same methods as global model aggregation. SMA-based policy validation mitigates model poisoning threats, while LFPN aggregates global models. Finally, MDCS and DEdS deliver the securely aggregated supermodel to cloud-IoT users and securely store it in the cloud database as represented in Fig 4.

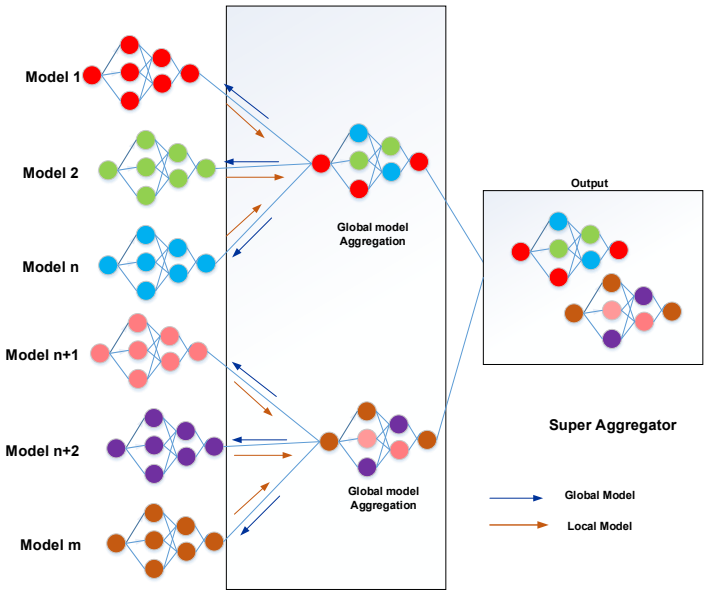

Fig. 4 Architecture of LFPN

c. Super Model Aggregation & Secure Distribution

The models from the MDCS are securely aggregated by the SA to improve the effective data sharing, predictive maintenance, and data sharing capability of the multi-cloud environment. Similar steps are taken place in the super model aggregation like global model aggregation. The model poisoning attacks are mitigated by SMA-based policy validation, and aggregation of the global model is performed by LFPN are indicates in Fig 4. Finally, the securely aggregated supermodel is provided to the cloud-IoT users via MDCS, and DEdS, and also securely stored in the cloud database.

III. EXPERIMENTAL RESULTS

This section presents the cloud-IoT ecosystem's end-to-end security and privacy protecting servers and user data. This experimental study is divided into three subsections: the simulation setup, comparison analysis, and research summary. The results section shows how the suggested work performs better than previous efforts.

A. Experimental Setup

The NS-3.26 network simulator's application of the simulation findings of the suggested work improves the research's performance. It was determined that our work performed better than the suggested framework when compared to several performance metrics. The system configuration is covered in Table (I), and the network parameter setting is covered in Table (II).

TABLE I: SYSTEM PARAMETERS

Hardware configuration	Hard disk	62 GB
	RAM	4GB
	Processor	CPU: Intel(R) Core (TM) i5-4590S @ 3.00 GHz

Software configuration	Network simulator	NS-3.26
	Operating system	Ubuntu 14.04 LTS

TABLE II: SIMULATION PARAMETER

Parameters		**Descriptions**
Network Parameters	No. of cloud-IoT users	50
	No. of TAS	1
	No. of DEds	5
	No. of SMA	5
	No. of MDCS	3
	Super aggregator	1
	Size of packets	100 bytes
	Simulation area	$1000m\ x\ 1000m$
Transmission Slot parameters	Slot length	1040 bits
	Slot duration	8μs
	packet length	830 bits
Packet Parameters	Packet Size	1024
	No. of. Packets	100 bytes
	No. of. Retransmission	Max 5
	Packet interval	0.99s
	Data rate	280kbps
Energy Parameters	Initial energy	0.5J
	Transmission power	47J
	Receiving power	47J
	Data aggregation power	5J
	Battery power	3.3V
No. of. Run		1100
Simulation time		150s
Probability of node		0.1

Number of rounds	600
Duration of a single round	18s

B. Comparative Analysis

The comparison analysis between the proposed framework and the current works is summarized in this section, where we take into account the previous works like Privacy Preserving Federated Learning (PPFL), Artificial Intelligent- Enabled Privacy Preserving (AI-EPP) [33], PMFL, Paillier Algorithm (PA), Federated Multicriteria Client Selection (FedMCCS) [34], Privacy-Preserving Convolutional Neural Network (PP-CNN) [35], Privacy Preserving Scheme (PPS) [36]. The proposed work achieved better performance in terms of the Number of Cloud-IoT Users Vs Malicious Traffic Rate, Number of Epochs Vs Accuracy, Number of Communication Rounds Vs Accuracy, Size of the Data Vs HE encryption Time, Number of HE Operation Vs Noise Budget, Attack Rate Vs Privacy threats.

a. Number of Cloud-IoT Users Vs Malicious Traffic Rate

Currently, the following formula may be used to describe the idea of "Number of Cloud-IoT Users Vs. Malicious Traffic Rate" inside the initiative:

$$\mathfrak{m}=\frac{\vartheta}{\mho} \times 100 \tag{31}$$

$\mathfrak{m}$ is the malicious traffic rate;ϑ refers to the number of malicious users; $\mho$ indicates the total number of cloud-IoT users.

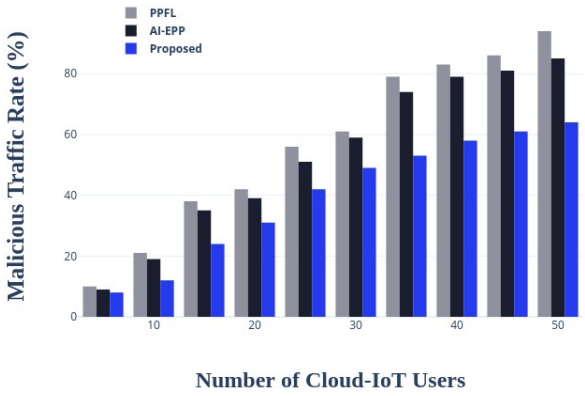

Fig. 5 Number of Cloud-IoT Users Vs Malicious Traffic Rate

Fig 5 illustrates the cloud-IoT users Vs malicious Traffic Rate. The comparison's results show that the recommended job performs better than similar works. The number of users increases when the malicious traffic rate also increases. 50 users with PPFL at 94%, AI-EPP at 85%, and finally proposed with 64%.

b. Number of Epochs Vs Accuracy

The relationship between the number of epochs and the trained models' accuracy is crucial in the context of the SMLFL system discussed in the research. The SMLFL framework includes several phases, including model aggregation and dissemination, user authentication, secure entity selection, and secure data exchange. To achieve robust and reliable federated learning outcomes, it is essential to understand how the number of training iterations (or "epochs") affects accuracy. $\mathcal{A}$ refers to the accuracy; $\mathfrak{S}$ indicates the number of epochs.

$$\mathcal{A} = f(\mathfrak{S}) \tag{32}$$

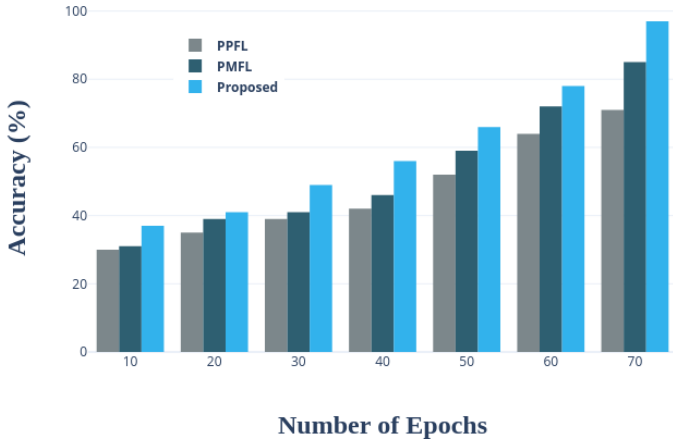

Fig. 6 Number of Epochs Vs Accuracy

Fig 6 indicates the number of epochs Vs accuracy. The comparative findings show that the recommended job performs better than previous efforts. Accuracy rises along with the number of epochs. In PPFL, there are 70 epochs with 71% accuracy, whereas in PMFL, there are 70 epochs with 85% accuracy and 97% proposal.

c. Number of Communication Rounds Vs Accuracy

In the secure federated learning architecture, the following simplified equation describes the relationship between the number of communication rounds (λ) and the accuracy ($\mathcal{A}$) of the aggregated model:

$$\mathcal{A} = \frac{\mathcal{K}}{(1+a\times\lambda)} \tag{33}$$

$\mathcal{K}$ is a scaling factor that establishes the highest level of accuracy possible; a is a constant that affects how quickly accuracy increases after each communication round.

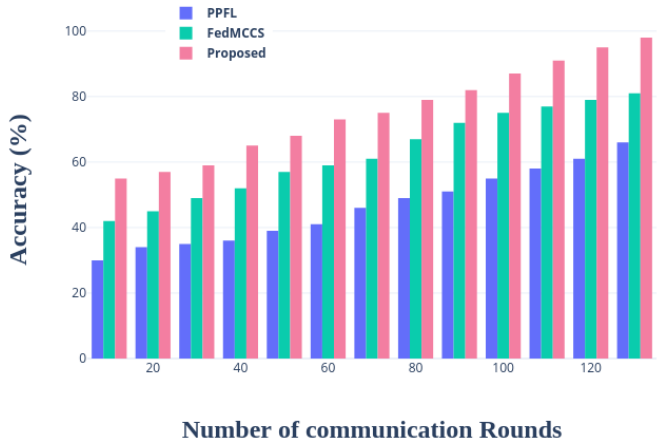

Fig.7 Number of Communication Rounds Vs Accuracy

Fig 7 represents the number of Communication Rounds Vs Accuracy. Communication rounds of 130 the PPFL's accuracy at 66, Communication rounds of 130 the FedMCCS's accuracy at 81, and proposed 98%. The findings of the comparison show that the proposed work performs better than the existing works.

d. Size of the Data Vs HE Encryption Time

The link between the size of the data (∂) and the length of time required for homomorphic encryption (ε) can be expressed using the following straightforward equation:

$$\varepsilon = \Psi \times \partial \tag{34}$$

Ψ is a constant that signifies the effectiveness of encryption.

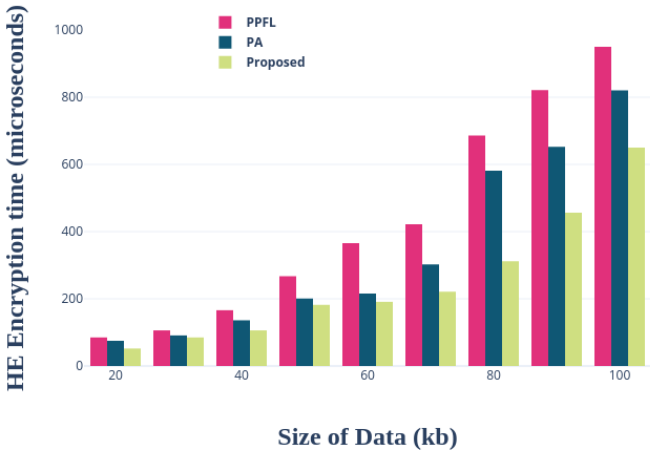

Fig.8 Size of the Data Vs HE Encryption Time

The size of the data Vs HE encryption time is illustrated in Fig. 8. PPFL size of data 100 with 950 μs, and in PA with 820 μs and finally proposed with 64 μs. The comparative findings demonstrate that the proposed work exceeds the current works.

e. Number of HE Operation Vs Noise Budget

The link between the number of homomorphic encryption (HE) operations (Π) and the available noise budget ($\eth$) can be expressed using the following straightforward equation:

$$\Pi = \mathfrak{A} \times \eth \tag{35}$$

$\mathfrak{A}$ shows the influence of each HE operation on the noise budget as a constant.

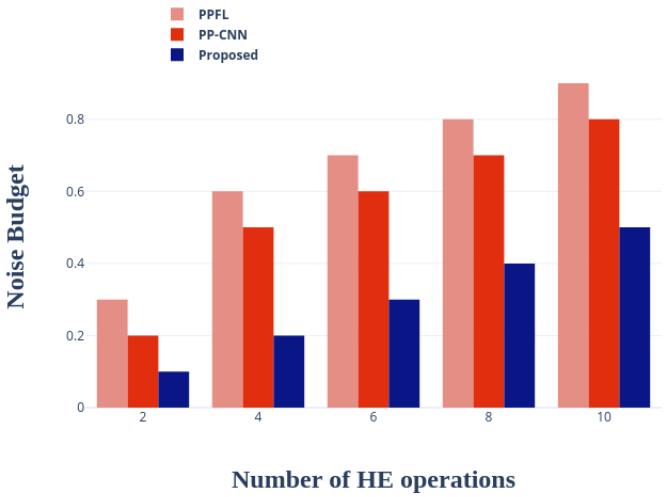

Fig. 10 Number of HE Operation Vs Noise Budget

The number of HE Operations Vs Noise Budget is shown in Fig. 10. Number of HE operations in 10 in PPEL with 0.9, PP-CNN with 0.8, and proposed with 0.5. The findings of the comparison show that the suggested work performs better when compared to current works.

f. Attack Rate Vs Privacy Threats

The relationship between the Attack Rate ($\Re$) and the Privacy Threats ($\mathfrak{y}$) can be expressed using the following straightforward equation:

$$\Re = \propto \times \mathfrak{y} \quad (35)$$

$\propto$ is a constant that depicts how each privacy threat affects the rate of attacks.

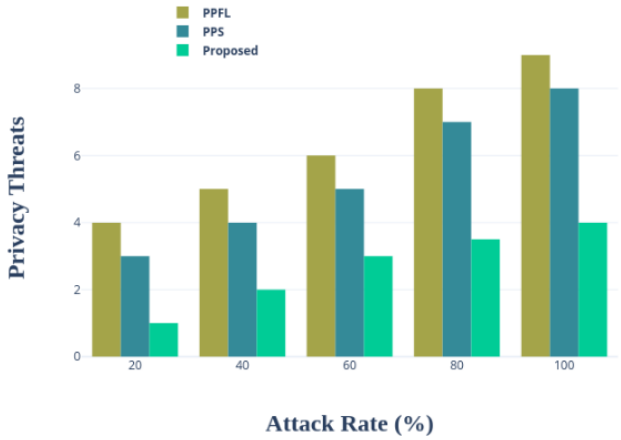

Fig.11 Attack Rate Vs Privacy Threats

Fig 11 indicates the attack rate Vs privacy threats. If the attack rate increases the privacy threats also increase. The attack rate is 100% the PPFL with 9, PPS with 8, proposed with 4. The findings of the comparison show that the proposed work outperforms existing works.

C. Research summary

In this section, we summarize the findings of our experiments, which demonstrate the improved performance achieved by the proposed framework. The tasks that will be completed as part of the proposed work are listed below: Number of Cloud-IoT Users Vs Malicious Traffic Rate, Number of Epochs Vs Accuracy, Number of Communication Rounds Vs Accuracy, Size of the Data Vs HE encryption Time, Number of HE Operation Vs Noise Budget, Attack Rate Vs Privacy threats which are described in fig. 5 to fig. 11. Table (III) shows the numerical analysis of new and existing works demonstrating the performance metrics.

TABLE III: PERFORMANCE ANALYSIS

Comparison metrics	PPFL	AI-EPP	PMFL	PA	FedMCCS	PP-CNN	PPS	Proposed
Number of Cloud-IoT Users Vs Malicious Traffic Rate (%)	94	85	-	-	-	-	-	64
Number of Epochs Vs Accuracy (%)	71	-	85	-	-	-	-	97
Number of Communication Rounds Vs Accuracy (%)	66	-	-	-	81	-	-	98

Size of the Data (kb) Vs HE encryption Time (μ)	950	-	-	820	-	-	-	650
Number of HE Operation Vs Noise Budget	0.9	-	-	-	-	0.8	-	0.5
Attack Rate Vs Privacy Threats	9	-	-	-	-	-	8	4

IV. CONCLUSION

In this research execute Three-Factor Novel Authentication to register cloud-IoT users to the TAS with their user ID, location, IP address, biometrics (finger and eye veins), and photo tag. The TAS issues AuthTok as proof of registration. Cloud-IoT users must pass three authentication levels. The TAS initially verified the user ID and password. Users are challenged by the TAS to choose a picture tag and provide concealed lines from their second-level registration. TAS asks the user to enter biometrics at the third stage. To create a pair of private and public keys for encryption and decryption for verified users, the TAS employed IKGP. Secure MLFL Entities Selection, TAS selects optimal DEdS utilizing the TEGT method. We selected clients using IANN. Next, Secure Privacy-Aware MLFL Homomorphic Data Sharing & Storage uses MLFL subprocesses such as local model, global model, and supermodel generation & aggregation. We employed HER-LresNet-EVO, a novel optimized deep learning-based homomorphic encryption approach. The global model aggregates normal models using lightweight deep learning called Lightweight Factorized Pyramidal Networks. Super Model Aggregation & Secure Distribution: This approach keeps supermodels in the cloud database and securely distributes them to cloud-IoT users via MDCS and DEdS. The suggested method is tested using NS-3.26, and its effectiveness is assessed by contrasting it with the strategies that are already in use. Numerical analysis is used to examine a method's performance, and it can be shown that our technique performs better than the current approaches across the board.

References

[1] Hassan, N. S., Abdulrahman, L. M., Delzy, M. M., Abdulkarim, N. M., Omar, M. A., & Sami, T. M. G. High-Performance Cloud Computing Services and its Influences by Web Technology Based on Information Systems.

[2] Aceto, G., Persico, V., & Pescapé, A. (2020). Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0. *Journal of Industrial Information Integration*, *18*, 100129.

[3] Jain, S., Gupta, S., Sreelakshmi, K. K., & Rodrigues, J. J. (2022). Fog computing in enabling 5G-driven emerging technologies for development of sustainable smart city infrastructures. *Cluster Computing*, 1-44.

[4] Mishra, S., & Tyagi, A. K. (2022). The role of machine learning techniques in internet of things-based cloud applications. *Artificial intelligence-based internet of things systems*, 105-135.

[5] Bagherzadeh, L., Shahinzadeh, H., Shayeghi, H., Dejamkhooy, A., Bayindir, R., & Iranpour, M. (2020, July). Integration of cloud computing and IoT (CloudIoT) in smart grids: Benefits, challenges, and solutions. In *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)* (pp. 1-8). IEEE.

[6] Gill, S. S., Tuli, S., Xu, M., Singh, I., Singh, K. V., Lindsay, D., ... & Garraghan, P. (2019). Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges. *Internet of Things*, *8*, 100118.

[7] Habib, G., Sharma, S., Ibrahim, S., Ahmad, I., Qureshi, S., & Ishfaq, M. (2022). Blockchain technology: benefits, challenges, applications, and integration of blockchain technology with cloud computing. *Future Internet*, *14*(11), 341.

[8] Patel, A., Shah, N., Ramoliya, D., & Nayak, A. (2020, November). A detailed review of cloud security: issues, threats & attacks. In *2020 4th International conference on electronics, communication and aerospace technology (ICECA)* (pp. 758-764). IEEE.

[9] Havanje, N. S., Kumar, K. R. A., Shenoy, S. N., Rao, A. S., & Thimmappayya, R. K. (2022). Secure and reliable data access control mechanism in multi-cloud environment with inter-server communication security. *Suranaree Journal of Science & Technology*, *29*(3).

[10] Megouache, L., Zitouni, A., & Djoudi, M. (2020). Ensuring user authentication and data integrity in multi-cloud environment. *Human-centric Computing and information sciences*, *10*, 1-20.

[11] Namasudra, S., Chakraborty, R., Kadry, S., Manogaran, G., & Rawal, B. S. (2021). FAST: Fast accessing scheme for data transmission in cloud computing. *Peer-to-Peer Networking and Applications*, *14*, 2430-2442.

[12] Seth, B., Dalal, S., Jaglan, V., Le, D. N., Mohan, S., & Srivastava, G. (2022). Integrating encryption techniques for secure data storage in the cloud. *Transactions on Emerging Telecommunications Technologies*, *33*(4), e4108.

[13] Ghanmi, H., Hajlaoui, N., Touati, H., Hadded, M., & Muhlethaler, P. (2022, March). A secure data storage in multi-cloud architecture using blowfish encryption algorithm. In *International Conference on Advanced Information Networking and Applications* (pp. 398-408). Cham: Springer International Publishing.

[14] Yin, L., Feng, J., Xun, H., Sun, Z., & Cheng, X. (2021). A privacy-preserving federated learning for multiparty data sharing in social IoTs. *IEEE Transactions on Network Science and Engineering*, *8*(3), 2706-2718.

[15] Alabdulatif, A., Khalil, I., Zomaya, A. Y., Tari, Z., & Yi, X. (2020). Fully homomorphic based privacy-preserving distributed expectation maximization on cloud. *IEEE Transactions on Parallel and Distributed Systems*, *31*(11), 2668-2681.

[16] Jiang, C., Xu, C., & Zhang, Y. (2021). PFLM: Privacy-preserving federated learning with membership proof. *Inf. Sci., 576*, 288-311.

[17] Fang, H.S., & Qian, Q. (2021). Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. *Future Internet, 13*, 94.

[18] Altaee, M.M., & Alanezi, M.K. (2021). Enhancing cloud computing security by paillier homomorphic encryption. *International Journal of Electrical and Computer Engineering, 11*, 1771-1779.
[19] Emad, S., Alanwar, A., Alkabani, Y., El-Kharashi, M.W., Sandberg, H., & Johansson, K.H. (2021). Privacy Guarantees for Cloud-based State Estimation using Partially Homomorphic Encryption. *2022 European Control Conference (ECC)*, 98-105.
[20] Ma, Z., Ma, J., Miao, Y., Liu, X., Choo, K.R., & Deng, R.H. (2020). Pocket Diagnosis: Secure Federated Learning Against Poisoning Attack in the Cloud. *IEEE Transactions on Services Computing, 15*, 3429-3442.
[21] Zhang, L., Zhang, Z., & Guan, C. (2021). Accelerating privacy-preserving momentum federated learning for industrial cyber-physical systems. *Complex & Intelligent Systems, 7*, 3289 - 3301.
[22] Yulliwas Ameur, Samia Bouzefrane, Thinh Le Vinh. Handling security issues by using homomorphic encryption in multi-cloud environment. The 14th International Conference on Ambient Systems, Networks and Technologies (ANT), Mar 2023, Leuven, Belgium
[23] Ren, W., Tong, X., Du, J., Wang, N., Li, S., Min, G., Zhao, Z., & Bashir, A.K. (2021). Privacy-preserving using homomorphic encryption in Mobile IoT systems. *Comput. Commun., 165*, 105-111.
[24] Fang, C., Guo, Y., Wang, N., & Ju, A. (2020). Highly efficient federated learning with strong privacy preservation in cloud computing. *Computers & Security*, *96*, 101889.

[25] Ahamad, D., Hameed, S. A., & Akhtar, M. (2022). A multi-objective privacy preservation model for cloud security using hybrid Jaya-based shark smell optimization. *Journal of King Saud University-Computer and Information Sciences*, *34*(6), 2343-2358.

[26] Ganapathy, S. (2019). A secured storage and privacy-preserving model using CRT for providing security on cloud and IoT-based applications. *Computer Networks*, *151*, 181-190.

[27] Pandiaraja, P., & Deepa, N. (2019). A novel data privacy-preserving protocol for multi-data users by using genetic algorithm. *Soft Computing*, *23*, 8539-8553.

[28] Li, S., Zhao, S., Min, G., Qi, L., & Liu, G. (2021). Lightweight privacy-preserving scheme using homomorphic encryption in industrial internet of things. *IEEE Internet of Things Journal*, *9*(16), 14542-14550.

[29] Park, J., & Lim, H. (2022). Privacy-Preserving Federated Learning Using Homomorphic Encryption. *Applied Sciences*.
[30] Boomija, M.D., & Raja, S.V. (2022). Securing medical data by role-based user policy with partially homomorphic encryption in AWS cloud. *Soft Computing, 27*, 559-568.
[31] Thabit, F., Can, O., Alhomdy, S., Al-Gaphari, G., & Jagtap, S.B. (2022). A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing. *International Journal of Intelligent Networks*.
[32] Kurniawan, H., & Mambo, M. (2022). Homomorphic Encryption-Based Federated Privacy Preservation for Deep Active Learning. *Entropy, 24*.
[33] Elhoseny, M., Haseeb, K., Shah, A. A., Ahmad, I., Jan, Z., & Alghamdi, M. I. (2021). IoT solution for AI-enabled PRIVACY-PREServing with big data transferring: an application for healthcare using blockchain. *Energies*, *14*(17), 5364.

[34] AbdulRahman, S., Tout, H., Mourad, A., & Talhi, C. (2020). FedMCCS: Multicriteria client selection model for optimal IoT federated learning. *IEEE Internet of Things Journal*, *8*(6), 4723-4735.

[35] Falcetta, A., & Roveri, M. (2022). Privacy-preserving deep learning with homomorphic encryption: An introduction. *IEEE Computational Intelligence Magazine*, *17*(3), 14-25.

[36] Zhao, P., Huang, H., Zhao, X., & Huang, D. (2020). P 3: Privacy-preserving scheme against poisoning attacks in mobile-edge computing. *IEEE Transactions on Computational Social Systems*, *7*(3), 818-826.