A secure federated learning privacy method for industrial IoT edge networks

Noname manuscript No.
(will be inserted by the editor)

A Secure Federated Learning Privacy Method for Industrial IoT Edge Networks

John Owoicho Odeh[1][*] · Yang Xiaolong[1][*] · Oluwarotimi Williams Samuel[2] · Sahraoui Dhelim[3] · Cosmas Ifeanyi Nwakanma[4]

Received: date / Accepted: date

Abstract The rapid growth of the Internet of Things (IoT) in industrial operations has driven the adoption of the Industrial Internet of Things (IIoT), necessitating intelligent networks of edge devices to efficiently generate, analyze, and utilize data from sensors. However, secure transmission of data within edge networks presents significant challenges, including privacy concerns and difficulties in secure data sharing. Existing methods addressing these issues often impose high computational overhead, negatively impacting efficiency. To address these limitations, a novel method, Federated Learning with Enhanced Privacy for Industrial IoT Edge Networks (FLEPNS), is proposed to adopt the edge network system and enhance privacy preservation while optimizing training efficiency. This approach incorporates the Paillier algorithm to implement an information masking mechanism and a shared token system, ensuring secure and obfuscated multi-device data sharing. FLEPNS achieves robust privacy protection without compromising model training accuracy or imposing substantial

John Owoicho Odeh
[1]Department of Computer and Communication Engineering, University of Science and Technology, Beijing 100083, China
E-mail: johnodeh@yahoo.co.uk

Xiaolong Yang
[1]Department of Computer and Communication Engineering, University of Science and Technology, Beijing 100083, China
E-mail: yangxl@ustb.edu.cn

Oluwarotimi Williams Samuel
[2] School of Computing and Data Science Research Centre, University of Derby, Derby, DE22 3AW, United Kingdom
E-mail: o.samuel@derby.ac.uk

Sahraoui Dhelim
[3] School of Computing, Dublin City University, Ireland
E-mail: sahraoui.dhelim@dcu.ie

Cosmas Ifeanyi Nwakanma
[4] ICT-Convergence Research Center, Kumoh National Institute of Technology, Gumi, 39177, South Korea
E-mail: cosmas.ifeanyi@kumoh.ac.kr

computational overhead. Additionally, a masking algorithm (SET) is introduced to counter adversarial attacks and ensure data integrity during sensor deployment and transmission between edge servers and devices. Experimental evaluations demonstrate that FLEPNS outperforms compared techniques for accuracy, showing a value of 62% for PAFLM and 70% for FLEPNS. For efficiency of privacy preservation, the FLEPNS has a higher value of 77% compared to 74%.Further evaluation reveals computational overhead and bandwidth usage by PALFM of 4.122MBps, in contrast to 3.1MBps for FLEPNS), showing significant advantage over compared techniques. These results highlight the distinct performance and practical benefits of FLEPNS in industrial edge network applications

Keywords Edge Network System · Federated learning (FL), Internet of Things, Industrial Internet of Things (IIoT), Privacy-preserving data analysis, and information masking

1 Introduction

The Industrial Internet of Things (IIoT) is an emerging technology that enables devices or sensors involved in industrial processes to communicate and interact with each other, making it possible to collect, process and analyze data in real time [19]. However, data collection and analysis within industrial IoT systems generate several challenges, particularly in terms of data privacy and security [28]. The use of machine learning algorithms in IIoT systems can help detect patterns and insights in data, but at the same time, it requires the collection of large amounts of data from various sources and the high level of accuracy usually begins to decline, as the big data reaches the threshold. Hence, federated learning (FL) [31] has emerged as a promising technique for training data sets and models on distributed data sources without compromising data privacy. Specifically, FL is a distributed machine learning technique that enables the training of learning models on decentralized devices without the need for centralized data storage. In other words, the centralized cloud only needs to collect the updated local training model from individual users in federated learning. This approach allows industrial IoT systems to train machine learning models on data generated by different edge devices while preserving the privacy of the data [18]. In addition, it also reduces the amount of data that needs to be transferred between devices, which can help overcome the limitations of network bandwidth and latency. Although federated learning is designed with privacy in mind, it may not be enough alone to guarantee complete privacy protection [21]. There are still risks associated with the data that is transmitted between devices or servers during the training process, and there may be ways for adversaries to infer sensitive information from the data. Privacy preservation techniques are therefore used to enhance federated learning to ensure that sensitive information is not compromised during the training and transmission process. Some common methods include differential privacy [7], Homomorphic encryption [32, 1], and blockchain [3], capable of supporting data privacy protection while enabling efficient training of machine learning models.

These strategies have been investigated in recent years as a privacy-preserving way to enhance federated learning for standard IIoT systems and IIoT-enabled edge

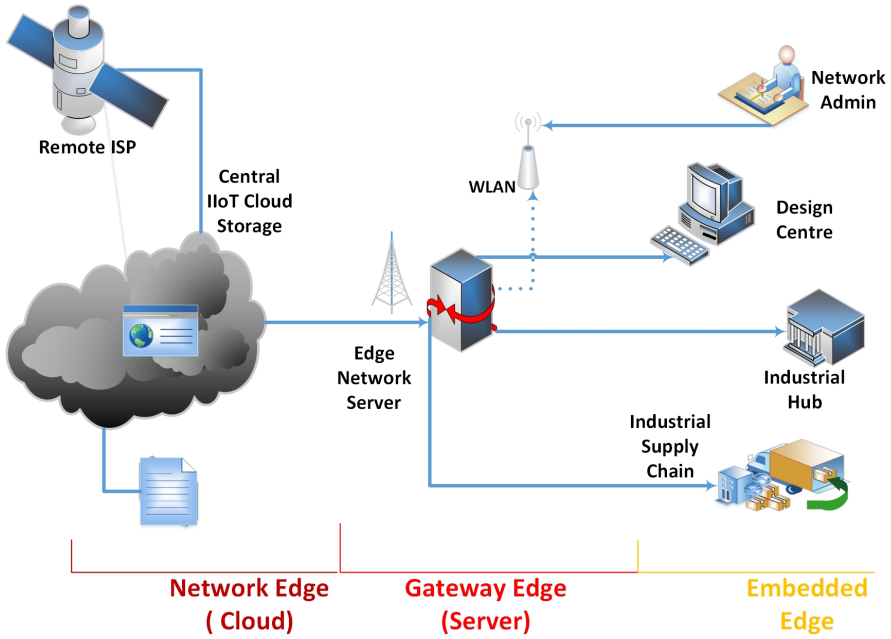

Fig. 1 : Industrial IoT enabled Edge Network System Architecture

network systems, as shown in Figures 1 and 2, respectively. This typical edge network-enabled IIoT system architecture has the following listed layers.

- Embedded Edge: This comprise of edge device such as a sensor, or laptop that is typically the source of the dataset, a locally trained model.
- Gateway Edge usually consists of the decentralized edge server that acts as the collector and aggregator of data from multiple peripherals or other gateways.
- Network Edge: This layer intermediates between the local network (embedded and edge gateway) and the extranet. The cloud facility is a perfect example of this layer [5]

The primary purpose of the edge network-enabled IIoT architecture is to exchange and ensure data privacy, integrity, and availability for actionable insights in an industrial application such as the Metaverse ecosystem[11]. For instance, an IIED 2, will supply data with a very high level of precision. This data will be useless until it is aggregated and locally trained, before being fed into the Metaverse database. This method makes it essential to safeguard the privacy and transmission security of the trained model[20].

Lu et al. [17] developed a secure blockchain-enabled data sharing architecture for multiple distributed devices. This method reveals only the data model, not the actual data. While Fu et al.[8], proposed verifiable federated learning with privacy-preserving method for big data in industrial IoT, Ruzafa-Alcázar et al [26] applied the differential privacy technique to the industrial IoT. This research was inspired by the realization that updated gradients and sizes in the training process also have the poten-

tial to compromise privacy. Another popular method of privacy preservation is Multi-Party Computation (MPC) [4], a cryptographic technique that allows multiple parties to jointly compute a function or perform a computation on their private data without revealing their inputs to each other. In industrial IoT (IIoT) applications, MPC can enable secure and privacy-preserving data sharing and analysis among multiple parties devices, such as sensors of manufacturers, suppliers, and customers, while preventing unauthorized access and data leakage.

However, ensuring the correctness of the results and addressing security and privacy concerns is challenging for MPC in IIoT. As a result, Olakanmi and Odeyemi [24] introduced a new secure offloading scheme that uses reputation and morphism for perfect verification of results and provides security requirements for effective MPC. Despite the successes of FL in IIoT, most of the existing privacy preservation methods often add extra computational costs. This is because these techniques require additional computations to be performed during the training process to ensure that the privacy of the participating devices' data is preserved. For example, to preserve privacy, the differential privacy-based approaches add noise to the training data to prevent an attacker from being able to infer on the individual data points. This noise addition requires additional computations during the training process, which can increase the overall computational cost. Similarly, encryption-based techniques require a centralized server and additional computational resources to encrypt and decrypt the data. Such additional costs can be a limiting factor for some organizations edge network, particularly those with limited computational resources. Thus, it is important to balance the privacy concerns with the computational costs to ensure that sensitive data remains protected while still allowing for effective machine learning [22].

Aiming to tackle the above mentioned challenge, this research proposes a federated learning-enabled information masking technique for Industrial IoT edge network systems (FLEPNS). This is designed to enhance privacy preservation while optimizing training efficiency, as the approach incorporates the Paillier algorithm to implement an information masking mechanism and a shared token system, ensuring secure and obfuscated multi-device data sharing. FLEPNS achieves robust privacy protection without compromising model training accuracy or imposing substantial computational overhead. Additionally, a novel "set" algorithm is introduced to counter adversarial attacks and ensure data integrity during sensor deployment and transmission between edge servers and devices. This research is highly significant as it addresses critical challenges in the rapidly evolving field of the Industrial Internet of Things (IIoT), as it represents a breakthrough by combining privacy preservation with optimized training efficiency and ensuring robust data security while maintaining high accuracy and minimal computational overhead.

The major contributions of this research are highlighted as follows;

- Propose a secure Federated Learning privacy method for Industrial IoT edge network systems (FLEPNS) to enhance privacy preservation and optimize training efficiency.
- The integration of the Paillier algorithm for information masking and shared tokens ensures secure data sharing while minimizing the computational burden

– A novel algorithm designed to safeguard data integrity against adversarial attacks during edge device operations
– Comprehensive Efficiency Metrics to demonstrates significant improvements in accuracy, efficiency, privacy preservation, and bandwidth usage compared to existing methods.

The rest of the paper is organized as follows; Section 2 presents a review of related works. Section 3 describes the secure Federated Learning privacy method for Industrial IoT edge network systems (FLEPNS). Experiments and analysis are displayed in Section 4. Section 5 concludes the paper, with a future research focus.

2 Related Works

Federated learning is a machine learning technique that enables the training of models on decentralized data, without requiring data to be collected and sent to a central server. This technique is particularly relevant in the context of Industrial IoT, where large amounts of sensitive data are generated and stored by edge devices, such as sensors and smart machines. In recent years, studies on privacy protection, [17]- [26], [14]-[34] have explored the potential of federated learning as a privacy-preserving solution for Industrial IoT. This section includes a review of several pertinent works in this field, as shown in Table 1.

One of the earliest works on federated learning in IIoT was performed by Koch et al.[14]. This study proposed a new approach called Federated Tensor Mining (FTM), which allows multiple nodes to share data in a secure way for tensor-based mining. FTM uses homomorphic encryption to enable mining from encrypted data, and several experiments show that FTM performs better than existing privacy-preserving methods. Specifically, FTM increases accuracy by up to 24 % compared to matrix-based privacy-preserving compressive sensing (PPCS) techniques. Zhang et al. [33] proposed privacy-preserving asynchronous deep learning methods, DeepPAR (privacy-preserving and asynchronous deep learning via re-encryption) and DeepDPA (dynamic privacy-preserving and asynchronous deep learning), which can protect each participant's input privacy and enable backward secrecy of group participants in a light-size manner. The proposed schemes are shown to be secure, efficient, and effective through security analysis and performance evaluations on real datasets. Arachchige et al. [2] proposed a framework called PriModChain that combines differential privacy, federated ML, Ethereum blockchain, and smart contracts to ensure privacy and trustworthiness in IIoT data. The feasibility of PriModChain was evaluated using simulations in Python and tested on local and public blockchain networks.

Furthermore, Liu et al. [16] proposed a new communication-efficient on-device federated learning (FL)-based deep anomaly detection framework for sensing time-series data in IIoT. The proposed framework uses an FL strategy to enable decentralized edge devices to collaboratively train an anomaly detection model, an attention mechanism-based convolutional neural network-long short-term memory (AMCNN-LSTM) model to accurately detect anomalies, and a gradient compression mechanism based on Top-k selection to improve communication efficiency. The proposed method was evaluated on four real-world datasets and it was shown to accurately and

Table 1 Related Works in Federated Learning and Privacy Preservation Approaches

Title	Year	Techniques/Approach	Key Contributions	Reference
Federated Tensor Mining (FTM)	2014	Homomorphic Encryption, Federated Learning	Secure data sharing via tensor mining, Improved accuracy compared to PPCS techniques	[[14]
DeepPAR and DeepDPA	2020	Re-encryption, Privacy-preserving asynchronous deep learning	Protects input privacy, Enables backward secrecy, Efficient and secure	[33]
PriModChain	2020	Differential Privacy, Federated Learning, Blockchain	Ensures privacy and trustworthiness in IIoT data, Tested on blockchain networks	[2]
FL-based Deep Anomaly Detection	2020	Federated Learning, AMCNN-LSTM model, Gradient compression	Communication-efficient anomaly detection, Reduced overhead by 50%	[16]
Verifiable Federated Learning (VFL)	2022	Lagrange Interpolation, Privacy-preserving mechanisms	Verifiable aggregated gradients, Blinding technology for privacy protection	[34]
Privacy-Preserving and Traceable FL (PPTFL)	2023	Hierarchical Aggregation, Blockchain	Traceable and tamper-proof model aggregation, Combats model tampering	[6]
Hierarchical Federated Learning	2020	Hierarchical Federated Learning	Enhances privacy through hierarchical aggregation, Challenges with scalability noted	[29]
RSA Algorithm for Data Encryption	2013	RSA Encryption	Basic data encryption and decryption in network environments	[9]
Hybrid Deep Learning Architecture	2020	Hybrid Deep Learning	Privacy-preserving mobile analytics	[25]
Intel Paillier Cryptosystem Library	2022	Paillier Cryptosystem	Homomorphic encryption for privacy preservation in IIoT	[13]
FLEPNS	2024	Information masking and token	Privacy preservation FL enabled Information masking and token	Our Work

timely detect anomalies while reducing the communication overhead by 50 %. A verifiable federated learning approach with privacy-preserving mechanisms called VFL was proposed by Fu et al.[34] to further address privacy issues in industrial IoT applications. VFL uses Lagrange interpolation to set interpolation points for verifying the correctness of aggregated gradients and employs blinding technology to protect privacy. The verification overhead of VFL remains constant regardless of the number of participants, and it guarantees that encrypted gradients of participants cannot be inverted by a malicious aggregation server with less than n-2 participants colluding. Experimental evaluations show that VFL performs well with high accuracy and efficiency.

Chen et al [6] considered that traditional federated learning may be vulnerable to model tampering, which may result in inaccurate models. Therefore, a Privacy-Preserving and Traceable Federated Learning (PPTFL) framework was proposed.

PPTFL consists of two main components: Hierarchical Aggregation Federated Learning (HAFL) [29] and blockchain-based model aggregation. HAFL is applied for privacy-preserving aggregation schemes to reduce the communication overhead and computation cost associated with traditional federated learning approaches. The second part combines federated learning with blockchain and IPFS to make the parameters traceable and tamper-proof [9]-[25]. Nonetheless, as the proposed approach uses HAFL, it may not be scalable for large-scale federated learning applications because, as the number of clients increases, the hierarchical structure becomes complex and the communication overhead also increases [13]. Also, the HAFL approach may lead to delays in training since it requires synchronization of the model updates across different levels, which can be challenging [22]. The proposed approach, described in the next section, is scalable, can simultaneously enhance privacy preservation, optimize training efficiency, and utilize less bandwidth to solve the above problem. It uses the added mask to obfuscate the trained dataset and the twin token to communicate among authorized edge devices. To the best of our knowledge within its scope, this is a unique research. Accordingly, all symbols, main notations, and their descriptions are displayed in Table 2.

Table 2 Symbols and Descriptions

Symbol	**Description**
$s(a,b,c)$	Sum of samples owned by IIED a, b, c. . . i.
IIED	Industrial IoT Edge Device
FLag	Aggregation gradient
θ_T	Training model shared
R	Round of training
BW	Channel Bandwidth
TD	Transmitted data
M_T	Transmission time
ESa	Element of each edge network server
IIEDC	Industrial IoT Edge Data Center
e	Error bit of the cross entropy of label and output
M_e	Transmission Energy
I	Iterations
T	Training time
DT	Local aggregation dataset
U	Gateway network set
LIG	Learning Integer Generator
TH	Limit/threshold
ϑ	Training output
φ	Learning rate
DIN	Data initialization node
K	Constant (of IID in aggregation area)
$\{N,g\}\{\lambda,\mu\}$	Public and private pair token

3 FLEPNS

This section describes the secure Federated Learning privacy method for Industrial IoT edge network systems (S-FLEPNS). The FLEPNS framework consists of three main components: a network system model, edge resource Optimization, and model aggregation that incorporates the information masking and unmasking technique.

3.1 Network System Model

This comprises a typical interconnected network system components represented in figure 2 and processes that make a functional edge infrastructure service[10] . The entities involved in this system include the Cloud Entity, edge network server, and edge devices [30]-[15]. The Cloud Entity is responsible for the connection to the internet for the upload, as well as updating of globally trained model during each training cycle. It is connected to the IIEDC (gateway), which is an intermediary between the cloud (storage) and the edge devices. Specifically, the IIEDC collects data from these smart terminals, aggregates them, and uses them to train the model. The resulting models are then uploaded to the cloud or downloaded to the edge devices. Finally, the Edge Network nodes or IIED are heterogeneous sensors or intelligent terminals that collect network and industrial data used for training. They play a crucial role in defining the different parameters and formulas used in creating the models.

Data aggregation on gateway edge network devices is a means of collecting, filtering, and approximating data from sensors and other embedded devices, for transmission through a gateway edge server to a remote network (cloud). This aggregation enables enhanced security and data privacy, as well as low latency, reduced bandwidth usage, and energy consumption. In this work, the distributed aggregation is utilized where the decentralized edge server, or IIEDC, selects a dataset from the IIEDs, then generates a mask, and public and private token to enable trained model secure transmission to the destination. This is achieved through federated learning and synchronization among the devices. The IIEDs gather the data and upload it to the gateway network device or the edge server (IIEDC) for dataset aggregation. These datasets are then used in the training of the model by the gateway network device. As a means of protection for the dataset, a mask is inserted, and the dataset is uploaded to the cloud for model aggregation. During this stage, the cloud unmasks aggregated modeled datasets and finally sends them back to the edge server for the next set of training. This process is then repeated until the global model converges. To represent this, we consider an edge network system S. For dataset and model aggregation, the edge system S, lets the allowed IIED devices a,b, and c respectively with parameters S(a,b,c), k be a constant, and A be the aggregation, allow( ) to define the constituent of the dataset authorized to participate in the model training as shown in equation (1) and in the information masking process, P(a,b,c).

$$S(a,b,c) = A.K.allow(IIEDa, IIEDb, IIEDc) \tag{1}$$

The privacy protection technique begins when the DIN activates the *A K get* function, similar to the Paillier cryptosystem algorithm [13]-[10] for the generation of pri-

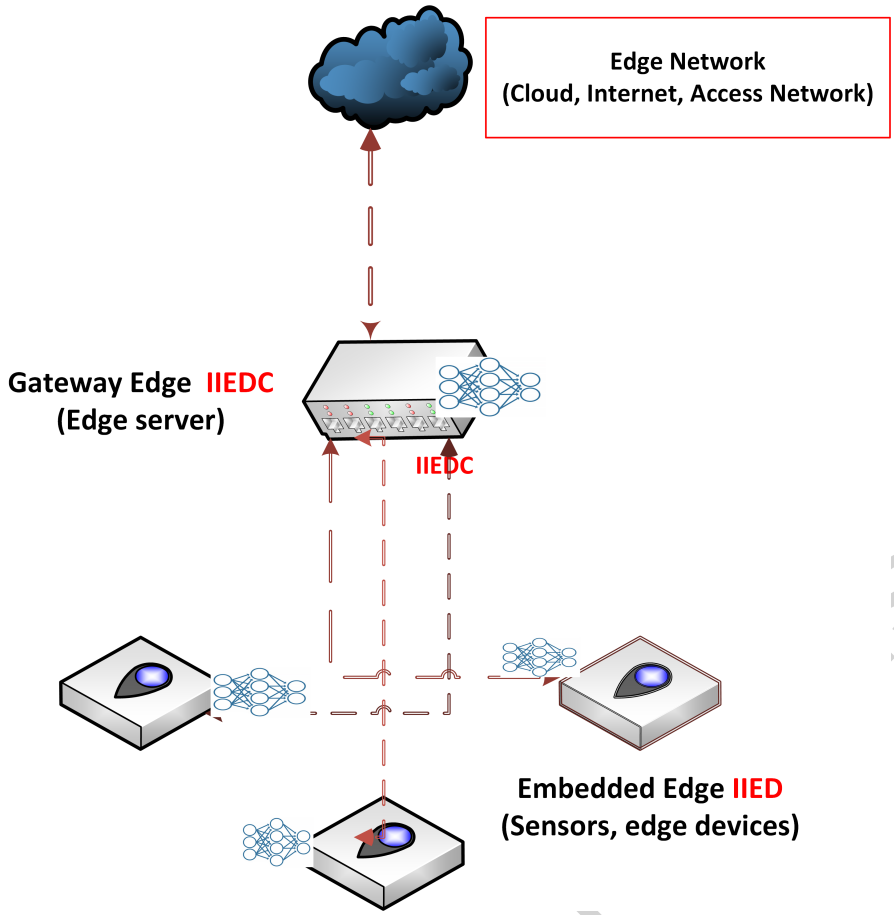

Fig. 2 Edge Network system

vate and public twin token (N, g) and (λ, μ) that are used for checking and verification of messages. The IIEDC produces random integer z, q, where z-1, and q-1 are prime numbers (set of bits more than 1024). To allow parameters in eqn. (1) to have protection parameters P, substituting eqn. (2) into the protection vector equation, thus; For a token generation, let p and q = two large prime numbers. These primes are used to generate the modulus n for the cryptosystem [30, 12].

Let lambda (λ): The Carmichael function, also known as the least common multiple of p − 1 and q − 1. It is used in various calculations within the Paillier cryptosystem [30], particularly for generating public and private tokens.

$$\lambda = LCM(z-1, q-1) \tag{2}$$

Let the plaintext message x be transformed using L(x) before being masked, set

$$L(x) = \frac{(x-1)}{N}, where, N = zq \tag{3}$$

To obfuscate the trained dataset model, select a token

$$g \in (Z * N^2) \tag{4}$$

Then, let g = LIG, the learning integer generator

Let the public and private pair token be (N, g) and (λ,μ)

Choose a random integer r from the set of invertible elements modulo N, denoted as Z*N, such that gcd(r, N) = 1

Let Pr = information mask

$$Pr = LIG^n * r^N modN + Sa(r^N modN) + Sb(r^N modN) + Sa(r^N modN). \tag{5}$$

where $\Delta = r^N$ mod N.

Let the sum of all random values and modulus = 1

$$Pr = LIG(IIED1,2,3) * \sum_{1}^{i} (r^N modN)(IIED1,2,3) IIED1,2,3 = \Delta a,b,c * LIG(IIED_1,2,3..n)) \tag{6}$$

Then the ciphertext mask results as follows:

$$Pr(a,b,c) = \Delta(a,b,c) * LIG\big(IIED(a,b,c)\big) \tag{7}$$

3.2 Edge Resource Optimization

Federated learning with dataset size can be calculated by the training time difference between when one edge device, uploads a trained dataset model (as compared to the sum of the sample of the learning node), and downloads corresponding dataset gradients [10]. It is known that federated learning represents a decentralized learning, scalable system, in which several edge devices contribute to model training after rounds of optimized updates. Here, the edge nodes request for download (black arrows) of the latest version of the parameter model from the edge server and update (brown arrows) the edge server with the latest round of gradient information as shown in Figure 3, this represents the parameter optimization and shows the gradients' staleness, which is parameter upload – gradient download process.

Therefore, the updating rules of the shared model are:

$$\theta = R^{i+1} \sum \theta(T) \tag{8}$$

where the amount of transmitted data TD in T training, having a channel bandwidth, BW; the model transmission time is given as:

$$M_T = \frac{TD}{BW} \tag{9}$$

For the energy used, where P is the transmission power of T training;

$$M_e = \frac{TD(P)}{BW} \tag{10}$$

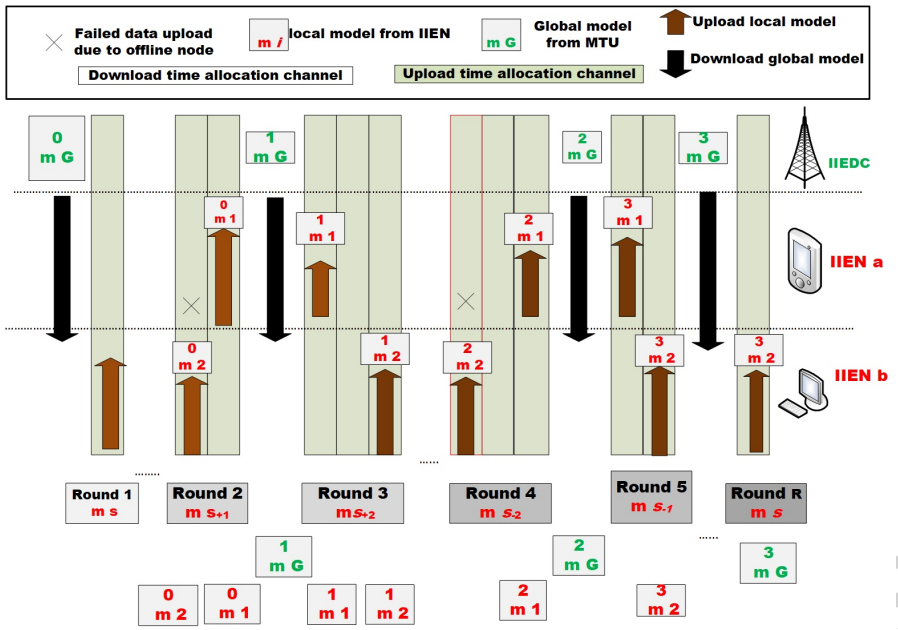

Fig. 3 Periodic Update Strategy[22]

To ensure the workablity of the process, some functions are deployed. The A.K.element () is used to secure constant k to generate related elements of the DH (Diffie-Hellman) token exchange protocol. SET( ) adopts the Paillier algorithm [25] to generate the private and public token twins. A.K.allow ( ) is based on the Paillier algorithm. It permits the edge parameters involved in utilizing the private and public tokens of the twin gateway. Unm( ) allows for the execution of a Secret reconstruction in the algorithm is based on the Lagrange interpolation [24]. It utilizes a connected private token to generate a secret share. TT.recon ( ) maximizes the use of shares for secret reconstruction, significantly reducing the computational complexity of secret reconstruction. ZIG.sign ( ) authorizes the public token twin with a private token for personal signature, and ZIG.verf ( ) verifies the signature public token set information. If the value of ZIG.verf (set information) = 1, it passes verification. We give a summary of each of them in Table 3.

3.3 Model Aggregation And Privacy Process

This section describes the federated learning process of the proposed FLEPNS.

3.3.1 The Initialization Stage:

Let *S (a, b, c . . . i) = Sa, Sb, Sc,. . . . Si* denote a set of IIED, having a dataset aggregated to smart edge devices or IIED. If at least L devices participate, one IIED is chosen as the dataset initialization node (DIN).

Table 3 Various Functions For Algorithm

Function()	**Meaning**
A.K.element()	Function to Public element generation in the algorithm
SET( )	Function to generate Private and public token twin in the algorithm
A.K.allow()	Function to permit Private shared token generation in the algorithm
TT.share()	Function to execute the Secret sharing algorithm
Unm( )	Function to execute a Secret reconstruction in the algorithm
ZIG.sign()	Function to activate timestamp and signature of the public token in the algorithm
ZIG.verf()	Verification of the token in the algorithm
End	End the gradient generation processes in the algorithm

These nodes operate using the Paillier cryptosystem to get (q, LIG, e): G1 *G1 $\rightarrow$ G2. The public and private token pair Let the public and private pair token
(N, g) and (λ,μ)
g0 is an element of G1, g1 is an element of Z*N
Choose a random integer r from the set of invertible elements modulo N, denoted as Z*N, such that gcd(r,N) = 1. This process is carried out as follows:

- Acquisition of a channel, F for algorithm sharing;
- Opening of a transmission channel for inter-level communication;
- Get a value n and TH limit for secret sharing protocol.
- Lastly, the gateway (IIEDC) utilizes z to generate parameters for the DH token exchange protocol.This exchange protocol is used by the z prime number to acquire data from the edge network devices through the transmission channel, F to the gateway device for storage.

3.3.2 Data Collection and Training Stage

The recurrent neural network is used as the training model. The calculation of the gradient depends on the data output of the previous step and the forecasts of the future output of the learning process. Although the input and the output are not related, the RNN shares the standard elements or parameters at every layer of the training process. With a learning rate of φ , the training process uses the bidirectional phases; straightforward and backpropagation. The straightforward phase computes the output of the layers and checks any error bit, e of the cross entropy of label and output. The backpropagation computes the gradient of each parameter as related to e and;

$$\vartheta = \vartheta - \frac{\varphi \delta e}{\delta \vartheta} \tag{11}$$

After this set of training iterations, each gateway will send an updated dataset to the cloud. The trained local model at the gateway will have an update of;

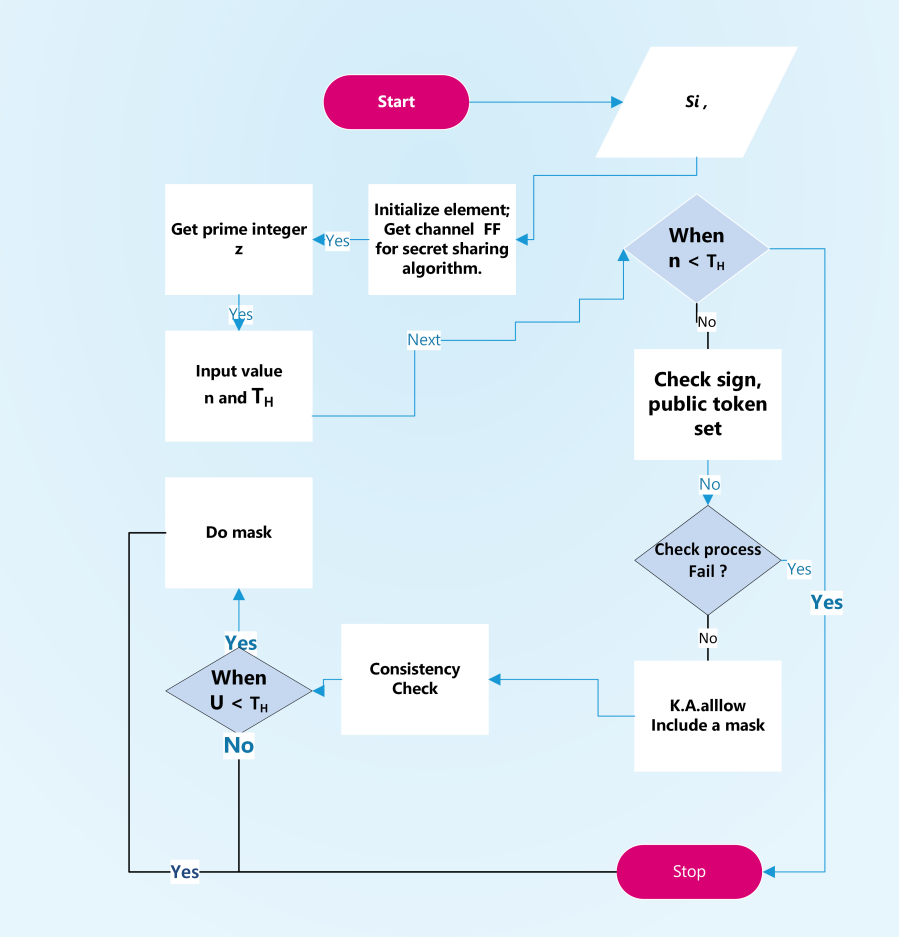

Fig. 4 Privacy Preservation Masking Flowchart

$$\theta(T,I+1) = \theta(T,I) - ng(\theta(T,I)) \tag{12}$$

This trained model is then uploaded to the cloud. The data output becomes the predicted output of;

$$\theta_{I+1} = \frac{1}{R}\sum \theta_T \tag{13}$$

The various processes and resource utilization are represented in a flow chart as shown in Figure 4.

3.3.3 The Protection Stage

Since an adversary can attack the dataset or trained model by through eavesdropping during transmission of the model, or change of model size, the "get-mask, *Pr( )*"

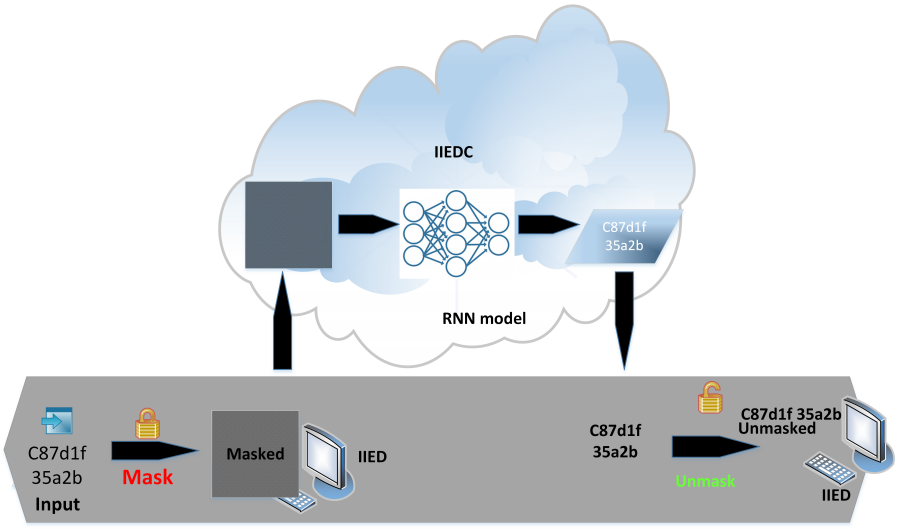

Fig. 5 Federated Learning Masking technique for FLEPNS

function is designed as a protection algorithm to input a mask to obfuscate the data set being transmitted. To obfuscate, the algorithm inserts a mask, adds tokens to the aggregated dataset, and makes it semantically the same as the original dataset [31], but less readable or hard to decipher by any adversary (as shown in Figure 5). It is important to note that this drastically hides the file so that data upload/ download between the edge server and IIED will be fast and secured[23]. This method holds for preservation as it concerns the use of three S.ID variables, a,b, and c, as recalled in the equation. (1) and (4) respectively.

$$S(a,b,c) = A.K.allow(IIEDa, IIEDb, IIEDc) \tag{14}$$

$$Pr_{a,b,c} = g \cdot \Delta(IIED_a + IIED_b + IIED_c) \tag{15}$$

Hence, the insertion of IIED, mask and Unmask parameter in the algorithm 1.

The “locate-size” function from the masking matrix algorithm gives an integer matrix that has a similar measurement as the size matrix of the model by calling the *interger.rand* () in the package. The simple act of masking ensures that the model size being delivered by the gateway (IIEDC) is well protected from malicious attacks and involves little or no computational burden, as compared to other protection techniques.

3.3.4 Aggregated Model and Unmasking Stage

When the authentication signature information of the model received by the cloud is confirmed, the cloud server unmasks the public and private masks to get the aggregated trained model and secures the updated model. After this complete model

Algorithm 1 Masking Algorithm

```
function SET(mine, sharedtoken, IIED)
    mine.token = sharedtoken
    mine.ID = IIED
    size = GET.COPY(mine.size)
    for each sid in sharedtoken do
        if S.ID ≥ IIED then
            size = ORDERDICT.TENSOR.INCLUDE(size, mine.shift.part.s, (sharedtokens[S.ID]^m ine.secrettoken)%mine.mod)
        else
            size = ORDERDICT.TENSOR(size, mine.shift.part.s, (sharedtokens[S.ID]^m ine.secrettoken)%mine.model)
        end if
        Check consistency = A.K.allow
        if U ¡ TH then
            Do Mask = Pr(a,b,c) = Δ(a,b,c) ∗ LIG(IIED(a,b,c)) Output model training at Source
        end if
        Check consistency = A.K.allow
        Undo Mask = Unm(Pr(a,b,c)) = L(g∗Δ(Sa,+Sb+Sc))^( − 1) Output model training at destination
        terminate process
    end for
end function
```

training process, the aggregated model is transmitted to the respective edge network devices (IIED), where the information unmasking process using the private token as represented in eqn. (11). The information is updated and prepared for the next round of model training.

To unmask the aggregated dataset from the IIEDC, the IIED makes use of the private token.

$$Unmask = L(P^{\lambda} mod N^2) * \mu mod N From \mu = (L(LIG^{\lambda} mod N^2))_1 mod N \tag{16}$$

$$Unm(Pr(a,b,c)) = L\big(g*\Delta(Sa,+Sb+Sc)\big)-^1 \tag{17}$$

4 Analytical Procedure And Experiment

In this section, is a description of the configuration settings is provided, which includes the data sets and the compared methods. The results of the experiments are then presented, along with the analysis drawn from them.

4.1 Experimental Setting

EdgeCloudSim v4.0 [27] is used to depict the IIoT system enabled by the edge network adopting the Hierarchical Edge Computing architecture as shown in Figure 2, as well as the basic Python language for testing purposes on the Windows 10 operating system. To evaluate the effectiveness of the proposed method, experiments are performed using the MNIST dataset. The MNIST dataset is a huge database of [1, 0]

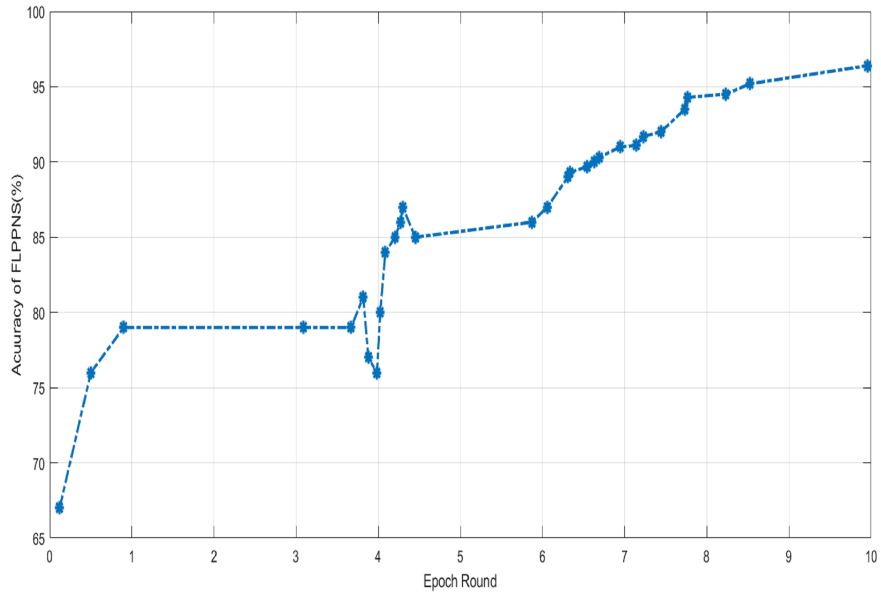

Fig. 6 Accuracy of FLEPNS

digits that is used as a benchmark for image classification tasks in machine learning. It has 12,000 images of [0, 1] digits, with 7,000 images used for model training and 5,000 images used for sampling. Each image is 28 pixels wide and 28 pixels high, and is grayscale, with each pixel having a value between 0 and 255 indicating the intensity of the gray color. The digits in the images range from 0 to 9, with each digit appearing roughly the same number of times.

4.2 Results And Analysis

In this subsection,is a comprehensive analysis of the results obtained from experiments conducted and the methods used to perform each experiment. An instance, is the accuracy level of FLEPNS as seen in Figure 6, then a comparison between the performance of the proposed FLEPNS with that of PAFLM and the Differential Privacy methods. These evaluation metrics focused on, in these comparison include; compute overhead, bandwidth utilization, privacy preservation assessment, performance accuracy, and the operational efficiency of the edge network system's data processing. Furthermore, while keeping track of time used in training , bandwidth consumption, and number of iterations or rounds (epoch), we investigate federated learning approaches for edge network intelligence with privacy protection in edge network systems.

4.2.1 Comparison of Accuracy

The level of FLEPNS accuracy of model training, measured in Table 4 is highlighted in Figure 7, which depicts the outcomes of the other methods, after applying the federated learning privacy preservation technique on the MNIST dataset. It was observed that during the initial stages of training, PAFLM achieves an accuracy of 62% while FLEPNS obtains an accuracy of 70%. Also, we noticed that when the learning rate is high the training passes the steepest point of the gradient descent process. And by the 18th iteration, the accuracy had improved significantly, reaching approximately.

Table 4 Comparison of Accuracy of FLEPNS,DP, PPAFLM Using MNIST

FLEPNS		Differential Privacy		PPAFLM	
Epoch	Accuracy(%)	Epoch	Accuracy(%)	Epoch	Accuracy(%)
0.12	67	4.693	11.1	0.15	66
0.5	76	4.761	11.5	0.8	72
0.9	79	5.181	11.7	3.4	73
3.087	79	5.195	14.5	3.8	73
3.669	79	5.917	16.6	3.95	76
3.816	81	6.01	17.9	3.99	78
3.879	77	6.446	18.2	4	76
3.983	76	6.547	22.4	3.991	75
4.023	80	6.788	25.9	4.2	78
4.086	84	7.175	28.4	4.3	81
4.202	85	7.51	29.4	4.4	86
4.272	86	7.673	33.7	4.2	86
4.296	87	7.712	37.2	4.6	86
4.451	85	7.952	40.1	4.5	87
5.87	86	11.484	43.4	6.1	93
6.058	87	11.571	45.7	6.3	95
6.308	89	12.167	44	6.34	95
6.334	89.3	12.313	45.1	6.454	96
6.543	89.7	12.625	47.3	6	96
6.627	90	14.696	51.3	5.8	96
6.689	90.3	14.856	55.6	9.846	89.3
6.945	91	15.324	60.1	10.918	90
7.138	91.1	16.262	63	11.557	90.1
7.229	91.7	17.312	66.7	11.585	90.7
7.44	92	18	68.9	11.931	92
7.731	93.5	18.4	70.2	12.336	92.7
7.766	94.3	18.9	73.1	12.624	93.2
8.227	94.5	19.5	71.4	15.086	94.3
8.527	95.2	19.6	70.6	16.843	95
9.954	96.4	19.9	70.4	19	95.6

4.2.2 Comparison for Efficiency

The efficiency of the systems is measured by assessing the time it takes to execute the model training. To evaluate the efficiency of the proposed method, it is compared

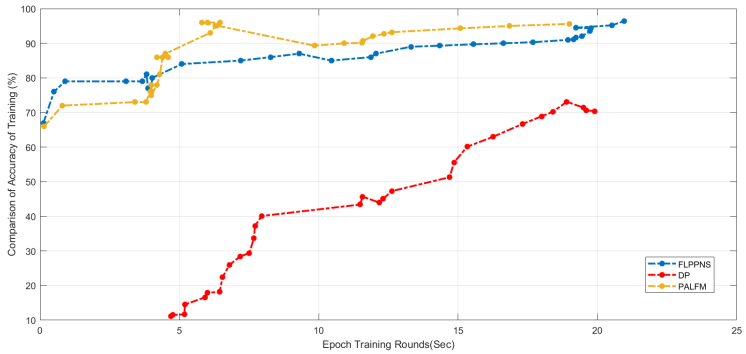

Fig. 7 Accuracy of FLEPNS,Differential Privacy and PALFM

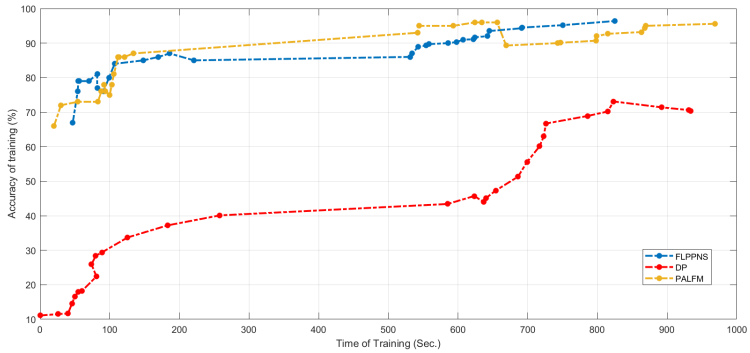

Fig. 8 Efficiency of FLEPNS, Differential, and PAFLM

with the PALFM algorithm and differential privacy, monitoring their performance when sampled under the same model training time T, as shown in Table 5.

As can be measured in Table 5 and displayed in Figure 8, the efficiency rate of the proposed FLEPNS is infinitesimally higher during the starting training time, than PAFLM method, but higher 77% compared to 74% within the 800th to 900th seconds respectively. In addition, PALFM's privacy protection technique shows a risk of some degree of privacy exposure, and our FLEPNS shows its efficiency in handling privacy lapses in the edge network system.

Moreover, at the set and appropriate learning rate, it can be seen that the differential privacy method showed a decline at 600 seconds and 720 seconds, even as FLEPNS continued to increase until 890 seconds. Although FLEPNS's operational efficiency was slightly lower than that of PAFLM at the 510th and 620th seconds, it still outperformed major differential privacy schemes, providing an advantage in real-time model training. Based on these findings, it can be deduced that FLEPNS is better than other methods both in terms of achieving higher training efficiency and ensuring network and data privacy. Therefore, FLEPNS is a reliable option for ensur-

Table 5 Comparison of Efficiency of FLEPNS,DP, PPAFLM Using MNIST

FLEPNS		Differential Privacy		PPAFLM	
Training Time (s)	Efficiency (%)	Training Time (s)	Efficiency (%)	Training Time (s)	Efficiency (%)
47	67	0	11.1	20	66
54	76	25.9	11.5	30	72
55	79	40.2	11.7	54	73
56	79	45.7	14.5	83	73
70	79	49.9	16.6	88	76
82	81	55	17.9	92	78
82	77	60.1	18.2	94	76
91	76	80.9	22.4	100	75
99	80	73.9	25.9	103	78
107	84	79.9	28.4	106	81
148	85	89	29.4	112	86
170	86	125.6	33.7	113	86
186	87	182.9	37.2	121	86
221	85	258	40.1	134	87
531	86	585	43.4	542	93
534	87	623	45.7	544	95
543	89	637	44	593	95
554	89.3	640	45.1	624	96
558	89.7	654	47.3	634	96
586	90	686	51.3	656	96
598	90.3	699	55.6	669	89.3
607	91	717	60.1	743	90
622	91.1	723	63	747	90.1
624	91.7	726	66.7	798	90.7
642	92	786	68.9	799	92
645	93.5	815	70.2	815	92.7
691	94.3	823	73.1	863	93.2
692	94.5	892	71.4	868	94.3
750	95.2	931	70.6	869	95
825	96.4	934	70.4	969	95.6

ing efficient performance and protecting data privacy in such an unstable and insecure environment as an industrial IoT edge network or sensor.

4.2.3 Evaluation of Privacy Technique

Figure 9 displays the experiment's results to verify this privacy method's effectiveness. Here, it ensured that the masked IoT nodes that participated in the training process received dataset samples in batches of 500 sets per training, with a subsequent increase of 100% in volume. As observed from the results, it was discovered that as the dataset sample size increased to 5000 items being trained by approximately 5 connected edge gateways, the level of privacy protection enhancement ability became stronger, it protects 495 datasets per time from about 5 IIEDs. Conversely, when as much as 40 nodes were processing over 1000 items per training time (secs.), the level of protection decreased. For emphasis, integer mask were added to all participating node's data in the simulated analysis of our privacy preservation model. Based on the Diffie-Hellman(DH) token transfer protocol, the function A K.allow is used

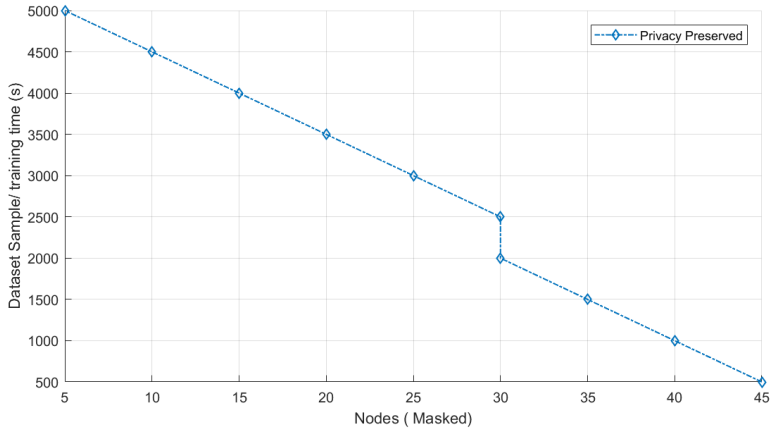

Fig. 9 Effectiveness of FLEPNS Privacy method

to get a random private token for each *Sa, Sb, Sc, Si.*, for the edge gateway server. This edge gateway device then makes the secret share through the secret sharing algorithm (TT.share ()) and shares it among the connected edge servers, while the obtained private token is used to get mask parameter Pr(a, b, c). This enables the edge gateways to generate personal masks and add the public mask that would ensure that the global model gradient is protected from exposure during transmission and upload to the cloud. In other words, our approach ensures that each edge gateway (IIEDC) only knows its dataset and its uploaded local model gradient; thus, keeping all other IIEDs' information secure. Though this has been simulated, the (TT.share ()) algorithm will prevent any would-be collision or jamming attack, while the ZIG.sign for timestamp, guarantees wholeness and confirmation of the uploaded information. Besides, the added mask ensures the security of model gradient data. Furthermore, even though some replay, man-in-the-middle, and other attacks might occur during training, upload, and transmission processes, the edge server will always be in contact with edge nodes that possess the token. Moreover, the timestamp will protect against such attacks.

4.2.4 Evaluation of Computational Overhead

In analyzing the computational overhead of FLEPNS, as compared to Differential privacy and the PALFM, where the dataset length = 100 the FLEPNS uses about 256 bits less computational overhead as compared to the other method. This results in increased communication effectiveness. Although it was discovered that the higher the number of IIEDs or the increase in the length of the dataset, the higher communication overhead in this method, however, when compared with other methods, it performs better as can be seen in Figure 10.

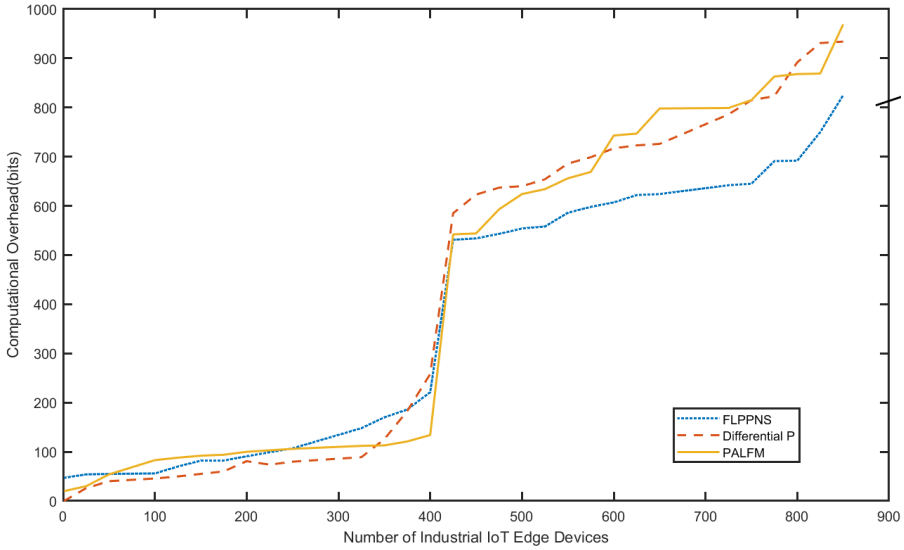

Fig. 10 Comparison of Communication Overhead

4.2.5 Evaluation of Bandwidth Usage

Lastly, in carrying out the model training as seen in Table 6 and the resultant outcome Figure 11, we discovered that the bandwidth usage of the FLEPNS is slightly lower during the different model upload and download, which in a way, has an impact of the model's accuracy. The outcome shows the resultant accuracy level during the federated learning epoch and the cost of bandwidth after 1000 epochs. The model configuration used the RNN b-directional and the MNIST dataset aggregated to IIEDs. For randomly selected IIEDs to train models, size is constant, using the backward regression where an accuracy of about 73% after 500 epochs is achieved by the PALFM using about 4.122MBps, in contrast to an accuracy of 81% using RNN after 500 epochs and 3.1MBps by our FLEPNS. This method utilizes less bandwidth as compared to the other evaluated methods.

5 Conclusion

The proposed FLEPNS framework introduces innovative techniques, including the integration of the Paillier algorithm and a shared token system, to ensure secure multi-device data sharing while maintaining high training accuracy and computational efficiency. Furthermore, the innovative masking algorithm (SET) greatly improves resilience against adversarial assaults and guarantees data integrity during transmission between edge servers and devices. Based on the novel algorithm, experimental results assessed for accuracy shows a value of 62% for PAFLM and 70% for FLEPNS. For efficiency of privacy preservation, the FLEPNS has a higher value of 77% compared to 74%.Further evaluation reveals computational overhead and bandwidth usage by

Table 6 Comparison of Bandwidth Usage

FLEPNS		Differential Privacy		PPAFLM				
Round	Accuracy (%)	BW (MB)	Round	Accuracy (%)	BW (MB)	Round	Accuracy (%)	BW (MB)
0	67	0.15	0	28.4	0.12	0	66	0.12
50	76	0.8	50	29.4	0.5	50	72	0.5
100	79	3.4	100	33.7	0.9	100	73	0.9
150	79	3.8	150	37.2	3.087	150	73	3.087
200	79	3.95	200	40.1	3.669	200	76	3.669
250	81	3.99	250	43.4	3.816	250	78	3.816
300	77	4	300	45.7	3.879	300	76	3.879
350	76	3.991	350	44	3.983	350	75	3.983
400	80	4.2	400	45.1	4.023	400	78	4.023
450	84	4.3	450	47.3	4.086	450	81	4.086
500	85	4.4	500	51.3	4.202	500	86	4.202
550	86	4.2	550	55.6	4.272	550	86	4.272
600	87	4.6	600	60.1	4.296	600	86	4.296
650	85	4.5	650	63	4.451	650	87	4.451
700	86	6.1	700	66.7	5.87	700	93	5.87
750	87	6.3	750	68.9	6.058	750	95	6.058
800	89	6.34	800	70.2	6.308	800	95	6.308
850	89.3	6.454	850	73.1	6.334	850	96	6.334
900	89.7	6	900	71.4	6.543	900	96	6.543
950	90	5.8	950	70.6	6.627	950	96	6.627
1000	90.3	6.1	1000	70.4	6.689	1000	89.3	6.689

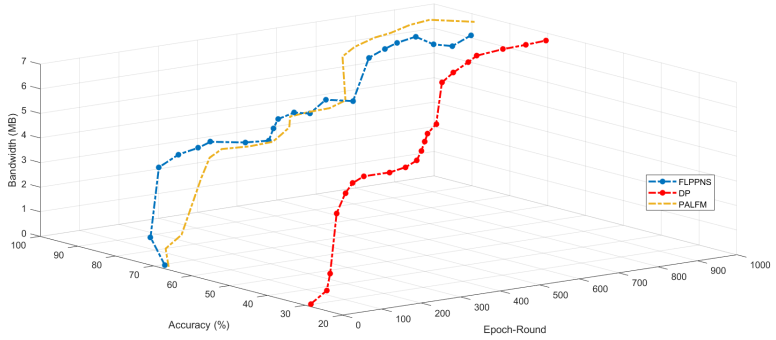

Fig. 11 Comparison of Bandwidth Transmission Usage

PALFM of 4.122MBps, in contrast to 3.1MBps for FLEPNS), showing significant advantage over compared techniques. Overall, this work demonstrates the effectiveness of FLEPNS in addressing privacy and efficiency challenges in the operation and deployment of the industrial Internet of Things within edge network systems.

6 Future Work

Further research work would be carried out in these areas;

6.1 Scalability and Security Measures:

Develop model synchronization techniques to minimize latency maximize efficiency and handle large-scale deployments with a high number of edge devices and participants. Also, enhanced Security Measures with advanced cryptographic techniques to fortify data privacy. This could involve exploring post-quantum cryptography, and improved key management systems to protect against emerging threats. This will strengthen defenses against adversarial attacks aimed at compromising data integrity or privacy within federated learning frameworks. Design of privacy-preserving Machine Learning Algorithms that are more efficient and compatible with federated learning settings. This could involve exploring differential privacy enhancements, federated learning optimizations for specific types of data (e.g., time series, image data), and federated transfer learning techniques.

6.2 Standardization and Framework Development:

The development of standards and frameworks for federated learning in industrial IoT environments, which includes norms for interoperability, data governance, and ethical concerns about data protection and utilization in federated learning environments. In addition, designing frameworks to address ethical and legal difficulties associated with data ownership, consent management, and regulatory compliance in federated learning applications. This includes working with service providers, legal experts, and others to ensure data is used responsibly and transparently.

6.3 Real-World Deployment and Validation:

Extensive validation and testing in real-world Industrial IoT settings is required to evaluate the efficacy and scalability of federated learning techniques. This includes working with industry partners to launch experimental initiatives, measure performance metrics, and collect feedback for continual improvement. Further research should include user-friendly interfaces and tools to create simple interfaces, toolkits, and platforms for deploying and managing federated learning models in edge computing environments, as this will enhance the integration of existing IoT frameworks and cloud services to provide efficient data flow and administration. Focusing on these aspects, future research in federated learning-enabled privacy preservation for industrial IoT edge network systems can considerably improve the field, making it more safe, efficient, and accessible for a wide range of applications.

– Acknowledgement:
This work is supported by the National Natural Science Foundation of China (NSFC) under Grants 61971033 and 61941113 and the Federal Scholarship Board, Ministry of Education, Abuja
– Conflict of interest/Competing interests: There is no conflict of interest to declare
– Author contribution:

Conceptualization, J.O.O., X.Y.; methodology, J.O.O.; software, J.O.O., X.Y.; validation, J.O.O., X.Y.; formal analysis, J.O.O.; investigation, J.O.O., X.Y., C.I.N., S.D. O.S.; data curation, J.O.O.; original draft preparation, J.O.O., X.Y., C.I.N.; writing, review and editing, J.O.O., X.Y., C.I.N., S.D.; visualization, J.O.O., X.Y., C.I.N.; supervision, X.Y. All authors have read and agreed to the published version of the manuscript.

References

[1] Abbas Acar et al. "A Survey on Homomorphic Encryption Schemes". In: *ACM Computing Surveys* 51 (Sept. 2018), pp. 1–35. DOI: 10.1145/3214303.

[2] Pathum Chamikara Mahawaga Arachchige et al. "A Trustworthy Privacy Preserving Framework for Machine Learning in Industrial IoT Systems". In: *IEEE Transactions on Industrial Informatics* 16 (Sept. 2020), pp. 6092–6102. DOI: 10.1109/tii.2020.2974555. (Visited on 12/17/2021).

[3] Abdullah Ayub Khan et al. "Internet of Things (IoT) Security With Blockchain Technology: A State-of-the-Art Review". In: *IEEE Access* 10 (2022), pp. 122679–122695. DOI: 10.1109/ACCESS.2022.3223370.

[4] Peter Bogetoft et al. "Secure Multiparty Computation Goes Live". In: *Financial Cryptography and Data Security* (2009), pp. 325–343. DOI: 10.1007/978-3-642-03549-4_20.

[5] Brandon Cannaday. "Hierarchical Edge Computing - A Practical Edge Architecture for IIoT". In: *LOSANT blog*. www.losant.com, 2020. URL: https://www.losant.com/blog/hierarchical-edge-computing-a-practical-edge-architecture-for-iiot (visited on 03/24/2024).

[6] Junbao Chen et al. "Privacy-Preserving and Traceable Federated Learning for data sharing in industrial IoT applications". In: *Expert Systems with Applications* 213 (Mar. 2023), pp. 119036–119036. DOI: 10.1016/j.eswa.2022.119036. (Visited on 11/10/2023).

[7] Cynthia Dwork. "Differential Privacy". In: *Automata, Languages and Programming* 4052 (2006), pp. 1–12. DOI: 10.1007/11787006_1.

[8] Anmin Fu et al. "VFL: A Verifiable Federated Learning With Privacy-Preserving for Big Data in Industrial IoT". In: *IEEE Transactions on Industrial Informatics* 18 (May 2022), pp. 3316–3326. DOI: 10.1109/tii.2020.3036166. URL: https://arxiv.org/pdf/2007.13585.pdf (visited on 03/30/2023).

[9] Nentawe Goshwe. "Data Encryption and Decryption Using RSA Algorithm in a Network Environment". In: *IJCSNS International Journal of Computer Science and Network Security* 13 (2013), p. 9. URL: http://paper.ijcsns.org/07%5C_book/201307/20130702.pdf.

[10] Fan Hongbin and Zhou Zhi. "Privacy-Preserving Data Aggregation Scheme Based on Federated Learning for IIoT". In: *Mathematics* 11 (Jan. 2023), p. 214. DOI: 10.3390/math11010214.

[11] Odeh JO and Yang X. "Industrial IoT Based Digital Twin and Metaverse Privacy and Security measures for A Trustworthy Industrial Metaverse Ecosystem". In: *IT Professional* 26 (6 2024). DOI: 10.1109/MITP.2024.3486115.

[12] Mirza Golam Kibria et al. *Big Data Analytics, Machine Learning and Artificial Intelligence in Next-Generation Wireless Networks*. arXiv.org, 2017. URL: https://arxiv.org/abs/1711.10089 (visited on 11/10/2019).

[13] Sejun Kim et al. *Intel Paillier Cryptosystem Library*. Intel, July 2022. URL: https://www.intel.com/content/www/us/en/developer/articles/technical/homomorphic-encryption/iso-compliant-paillier-cryptosystem-library.html.

[14] Robert Koch, Mario Golling, and Gabi Dreo Rodosek. "Behavior-based intrusion detection in encrypted environments". In: *IEEE Communications Magazine* 52 (July 2014), pp. 124–131. DOI: 10.1109/mcom.2014.6852093. (Visited on 03/28/2020).

[15] Xiang Liu. "Optical Communications in the 5G Era". In: *Elsevier eBooks* 978-0-12-821627-9 (Jan. 2022). DOI: 10.1016/c2019-0-03956-8. (Visited on 03/25/2024).

[16] Yi Liu et al. "Deep Anomaly Detection for Time-series Data in Industrial IoT: A Communication-Efficient On-device Federated Learning Approach". In: *IEEE Internet of Things Journal* 8.8 (2021), pp. 6348–6358. DOI: 10.1109/jiot.2020.3011726. (Visited on 12/16/2020).

[17] Yunlong Lu et al. "Blockchain and Federated Learning for Privacy-preserved Data Sharing in Industrial IoT". In: *IEEE Transactions on Industrial Informatics* 16.6 (2020), pp. 4177–4186. DOI: 10.1109/tii.2019.2942190.

[18] Bheema Shanker Neyigapula. "Secure AI Model Sharing: A Cryptographic Approach for Encrypted Model Exchange". In: *International Journal of Artificial Intelligence and Machine Learning* 4.1 (Aug. 2024), pp. 48–60. DOI: 10.21203/rs.3.rs-3210577/v1. (Visited on 03/24/2024).

[19] D. C. Nguyen et al. "Federated Learning for COVID-19 Detection with Generative Adversarial Networks in Edge Cloud Computing". In: *IEEE Internet of Things Journal* 9.12 (2022), pp. 10257–10271. DOI: 10.1109/jiot.2021.3120998.

[20] John Nguyen et al. "Federated Learning with Buffered Asynchronous Aggregation". In: *arXiv:2106.06639 [cs]* (Mar. 2022). URL: https://arxiv.org/abs/2106.06639.

[21] Truc Nguyen and My T. Thai. "Preserving Privacy and Security in Federated Learning". In: *IEEE/ACM Transactions on Networking* 32.1 (2024), pp. 833–843. DOI: 10.1109/TNET.2023.3302016.

[22] John Owoicho Odeh et al. "Asynchronous Privacy-Preservation Federated Learning Method for Mobile Edge Network in Industrial Internet of Things Ecosystem". In: *Electronics* 13.9 (2024). ISSN: 2079-9292. DOI: 10.3390/electronics13091610. URL: https://www.mdpi.com/2079-9292/13/9/1610.

[23] John Owoicho Odeh et al. "Context Privacy Preservation for User Validation by Wireless Sensors in the Industrial Metaverse Access System". In: *Algorithms* 17 (May 2024), pp. 225–225. DOI: 10.3390/a17060225. (Visited on 07/09/2024).

[24] Oladayo Olufemi Olakanmi and Kehinde Oluwasesan Odeyemi. "Secure reputation and morphism-based offloading scheme: A veritable tool for multi-party computation in Industrial Internet of Things". In: *Concurrency and Computa-*

tion: Practice and Experience 34 (July 2022). DOI: 10.1002/cpe.7116. (Visited on 10/06/2022).

[25] Seyed Ali Osia et al. "A Hybrid Deep Learning Architecture for Privacy-Preserving Mobile Analytics". In: *IEEE Internet of Things Journal* 7 (2020), pp. 4505–4518. DOI: 10.1109/JIOT.2020.2967734. URL: https://ieeexplore.ieee.org/abstract/document/8962332 (visited on 05/11/2021).

[26] Pedro Ruzafa-Alcazar et al. "Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT". In: *IEEE Transactions on Industrial Informatics* 19 (Feb. 2023), pp. 1145–1154. DOI: 10.1109/tii.2021.3126728. (Visited on 10/31/2023).

[27] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. "EdgeCloudSim: An environment for performance evaluation of edge computing systems". In: *Transactions on Emerging Telecommunications Technologies* 29 (Aug. 2018), e3493. DOI: 10.1002/ett.3493.

[28] Sasu Tarkoma, Suliman Alghnam, and Michael D. Howell. "Fighting pandemics with digital epidemiology". In: *EClinicalMedicine* 26 (Sept. 2020), p. 100512. DOI: 10.1016/j.eclinm.2020.100512. (Visited on 10/15/2020).

[29] Aidmar Wainakh et al. "Enhancing Privacy via Hierarchical Federated Learning". In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (Sept. 2020). DOI: 10.1109/eurospw51379.2020.00053. (Visited on 06/26/2022).

[30] Ruijin Wang et al. "Privacy-Preserving Federated Learning for Internet of Medical Things under Edge Computing". In: *IEEE Journal of Biomedical and Health Informatics* 27 (2022), pp. 1–1. DOI: 10.1109/JBHI.2022.3157725. URL: https://ieeexplore.ieee.org/abstract/document/9729996 (visited on 12/05/2022).

[31] Qiang Yang et al. "Federated Machine Learning: Concepts and Applications". In: *ACM Transactions on Intelligent Systems and Technology* 10.2 (Feb. 2019), pp. 1–19. DOI: 10.1145/3298981.

[32] Xun Yi et al. "Single-Database Private Information Retrieval from Fully Homomorphic Encryption". In: *IEEE Transactions on Knowledge and Data Engineering* 25 (May 2013), pp. 1125–1134. DOI: 10.1109/tkde.2012.90. (Visited on 08/01/2021).

[33] Xiaoyu Zhang et al. "DeepPAR and DeepDPA: Privacy Preserving and Asynchronous Deep Learning for Industrial IoT". In: *IEEE Transactions on Industrial Informatics* 16 (Mar. 2020), pp. 2081–2090. DOI: 10.1109/tii.2019.2941244. (Visited on 01/29/2022).

[34] Bin Zhao et al. "Anonymous and Privacy-Preserving Federated Learning With Industrial Big Data". In: *IEEE Transactions on Industrial Informatics* 17 (Jan. 2021), pp. 6314–6323. DOI: 10.1109/tii.2021.3052183. (Visited on 05/26/2023).

Authors

John Owoicho Odeh obtained B.Sc. and M.Sc. degree in Computer Science. He is a Cisco Certified Network Associate and an IPv6 Certified Engineer. He is with the Internet Society as a Research Associate. He is currently doctoral researcher in Information and Communication Engineering at the University of Science and Technology, Beijing. He is a Senior Member, IEEE.

Xiaolong Yang received the B.Eng., M.S., and Ph.D. degrees in communication and information systems from University of Electronic Science and Technology of China, Chengdu, China, in 1993, 1996, and 2004, respectively. He is currently a Professor with the School of Computer and Communication Engineering, Institute of Advanced Networking Technologies and Services, University of Science and Technology Beijing, Beijing, China. He has fulfilled more than 30 research projects. He has authored more than 80 articles. His current research interests include the next-generation Internet, network security and defense, and anonymity networking. He holds 16 patents in these areas. He is a Member, IEEE.

Oluwarotimi Williams Samuel obtained a Ph.D. degree in Pattern Recognition and Intelligent Systems from the University of Chinese Academy of Sciences, Beijing, courtesy of the CAS-TWAS President's Fellowship with excellent dissertation honor and graduate awards. Before that, he received his bachelor's and master's degree in Computer Science with first-class honors and distinction, respectively. He is currently a Senior Lecturer with the School of Computing and Engineering, University of Derby, United Kingdom. His research focuses on developing cutting-edge solutions to core problems in Cyber-physical Systems (Rehabilitation robotics, Clinical decision support systems, Human-machine interfaces, etc.) and he is the principal/co-investigator of various reputable national and international research grants. He has a track record of 100+ articles in reputable peer-reviewed international journals and IEEE conference proceedings with several "Top Cited/Influential Papers" courtesy of Web of Science, ESI-Index, and IOP Science. Besides, his scholarly contributions have earned him several academic awards and honors. He has chaired, co-chaired, and served/serving as a Technical Program Committee Member of various IEEE international conferences.He is a Senior Member, IEEE.

Sahraoui Dhelim a senior postdoctoral researcher at University College Dublin, Ireland. He was a visiting researcher at Ulster University, UK (2020-2021). He obtained his PhD degree in Computer Science and Technology from the University of Science and Technology Beijing, China, in 2020 and a Master's degree in Networking and Distributed Systems from the University of Laghouat, Algeria, in 2014. He serves as a guest editor in several reputable journals, including Electronics journal and Applied Science Journal. His research interests include Social Computing, Smart Agriculture, Deep-learning, Recommendation Systems and Intelligent Transportation Systems.

Cosmas Ifeanyi Nwaknama is a Senior Research Fellow at the ICT-Convergence Research Center, Kumoh National Institute of Technology, South Korea. He received his Ph.D. in IT-convergence engineering from Kumoh National Institute of Technology, Gumi, South Korea, in 2022. He was an Intern with Asea Brown

Boveri, Lagos, Nigeria, in 2003. From 2006 to 2009, he was a Senior Banking Assistant with the First Bank of Nigeria PLC, Owerri, Nigeria. From 2009 to 2019, he was a Lecturer and a Researcher at the Federal University of Technology, Owerri, Nigeria. From 2019 to 2022, he was the Senior Research Assistant and laboratory manager at the Networked Systems Laboratory, School of Electronics Engineering, Kumoh National Institute of Technology, South Korea. His research interests include the reliability and explainability of artificial intelligence and the Internet of Things applications for smart factories, homes, farms, vehicles, and the metaverse. Dr. Nwakanma is a member of the Computer Professionals Registration Council of Nigeria (CPN), Institute of Electrical and Electronics Engineers (IEEE), Nigerian Society of Engineers, and registered by the Council for the Regulation of Engineering in Nigeria. He is a Guest Editor with the Future Internet and Discover Internet of Things journals.