

A Performance Optimized Privacy-Aware Edge Computing Architecture Using Selective Homomorphic Encryption and Adaptive Differential Privacy

Apurva Khandekar

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana - 500075, India

Department of Computer Science and Engineering, Gokaraju Rangraju Institute of Engineering and Technology, Hyderabad, Telangana - 500090, India

Email: khandekarapurva@gmail.com

ORCID iD: <https://orcid.org/0000-0002-6407-9868>

Prathipati Ratna Kumar*

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana - 500075, India

Email: rk30111972@klh.edu.in

ORCID iD: <https://orcid.org/0000-0003-2043-2743>

Received: February 5, 2026; Revised: April 10, 2026; Accepted: July 9, 2026; Published: October 8, 2026

Abstract: Edge computing has become a fundamental paradigm in real-time data processing of latency-sensitive applications like smart healthcare, Internet of Things (IoT), and financial systems. Nevertheless, current edge and cloud-based solutions do not provide a high level of privacy or have a high level of computational and communication overhead because of intensive cryptographic actions. In this paper, a lightweight privacy-conscious edge computing architecture is proposed and can be used to provide secure and low-latency computation on sensitive data based on a combination of feature-level selective lightweight homomorphic encryption, context-sensitive differential privacy, and fully edge-enforced privacy control. In contrast to traditional solutions, which use consistent encryption or fixed privacy controls, the proposed system only encrypts privacy-sensitive data characteristics and dynamically adjusts differential privacy noise depending on the actual system state. The design utilizes significantly fewer resources and offers superior scalability without compromising privacy guarantees. Extensive simulations in a realistic edge computing environment show that the proposed architecture can reduce the encryption overhead by up to 60 percent, reduce the communication latency to approximately 2 ms, and maintain an accuracy of over 98 percent under strict privacy requirements. The comparison of results indicates that the suggested framework is more efficient in computing, scalable, and compliant with regulations than cloud-based and current edge computing solutions. The suggested solution offers a realistic and scalable answer to real-time privacy-preserving edge analytics within resource-constrained environments.

Index Terms: Privacy-Aware Edge Computing, Selective Homomorphic Encryption, Differential Privacy, Secure Aggregation, Real-Time Analytics.

1. Introduction

The Edge computing paradigm has been proposed as one of the solutions to the drawbacks of the classical cloud-centric systems by allowing the processing of data nearer to the data sources. This paradigm is particularly important to the latency-sensitive and bandwidth-intensive applications, most notably, Internet of Things (IoT) networks, smart healthcare systems, autonomous vehicles, and Industry 4.0 settings [1,2]. By computing the data at the point of generation, edge computing reduces the delays in communication and reduces the burden imposed on centralized cloud computing systems. Nevertheless, with edge computing, which is now processing large amounts of sensitive data, privacy and security have become a significant challenge [3]. The areas of healthcare, financial services, and smart

infrastructure deal with very confidential user data, and the breach of data can result in identity theft, financial damages, and serious legal implications in these domains. In contrast to centralized cloud systems, which assume the presence of advanced security mechanisms, edge environments are decentralized in nature and resource-constrained, and the direct use of traditional privacy solutions is not feasible [4,5].

Although edge computing has benefits, it is not yet clear how privacy-preserving computation can be ensured to occur at the edge nodes. Although much has been done on privacy-preserving computation, the current methods have severe constraints in edge computing systems [6]. Cloud-centric solutions are based on processing and encryption in the center and transmission and storage, but need to decrypt the data to compute it, which makes the solution highly latency-prone and has a centralized trust. Homomorphic encryption-based models permit computation on encrypted data, but fully homomorphic encryption has prohibitive computational costs, and partially homomorphic schemes encrypt whole datasets and are not flexible to resource-constrained edge devices [7,8]. Differential privacy has been implemented in distributed systems and edge systems because it is computationally inexpensive, but the majority of implementations use fixed privacy budgets and do not ensure privacy during intermediate computation stages. Federated learning and secure multi-party computation eliminate the raw data sharing, but increase the communication overhead and are susceptible to inference attacks on model updates [9]. On top of this, hybrid edge-cloud architectures still require trusted cloud constituents, restricting privacy enforcement and regulatory compliance. Such constraints expose research gaps such as the lack of selective encryption schemes, the lack of adaptive privacy policy, massive computational and communication overhead, and the lack of support for real-time edge analytics [10]. To overcome these challenges, this paper presents a context-adaptive, edge-only privacy-sensitive architecture, which is able to combine selective lightweight homomorphic computation and dynamic differential privacy to provide strong privacy assurances with low latency and high computational efficiency in resource-constrained edge environments. Thus, there is an urgent requirement for lightweight, effective, and scalable privacy-saving mechanisms specially tailored to the context of decentralized edge computing.

The main goal of the proposed research is to develop an efficient and privacy-conscious edge computing architecture that is able to conduct safe computation on sensitive data at the edge. The proposed solution ensures the protection of the privacy of the users without undermining real-time performance or overloading resource-constrained edge devices. The architecture offers high-throughput processing of data and strong privacy guarantees by incorporating the use of selective encryption and adaptive privacy. The work is aimed at the practical implementation in the field of IoT systems, healthcare monitoring, and financial applications, where data privacy is a critical concern, and the latency is a strict requirement.

1.1. Contributions

In the current paper, several new contributions to privacy-aware edge computing are presented:

- Selective Lightweight Homomorphic Computation Framework Proposed Selective Lightweight Homomorphic Computation framework in which only sensitive data attributes are encrypted with lightweight homomorphic encryption, and non-sensitive attributes are computed in plaintext.
- Design a Dynamic Differentiated Privacy Noise Calibration algorithm, which dynamically scales noise by volume of data, frequency of aggregation, and workload of the edges.
- The proposed Edge-Only Privacy Enforcement Architecture provides no dependency on trusted cloud computation as it enforces all the privacy protection mechanisms at the edge.
- An original model of aggregation, the Lightweight Privacy-Preserving Aggregation Model for Real-Time Edge Analytics, is presented, which combines selective homomorphic computation and adaptive privacy noise.

2. Related Work

Mythili et al. [11] provide an innovative method to enhance the privacy of edge computing with the help of safe information transmission and deep learning optimization with the Internet of Things. Tusa et al. [12] suggest a marketplace that uses Paillier Homomorphic Encryption to optimize efficiency and protect client privacy while enabling intelligent function performance in a multi-provider scenario throughout the Edge-Cloud continuum. To reduce the pressure on server farms and other centralized computing devices, Sivash et al. [13] offer a new architectural model of cloud decentralization called hybrid edge cloud (HEC) that reduces the usage of network bandwidth and employs resources on smart devices. Abdulrahman et al. [14] developed a methodology to ensure its application in the resource-constrained setting of the healthcare system. It was constructed utilizing the Elliptic Curve Cryptography (ECC) with MD5 techniques. A specific signature forgery attack utilizing the underlying pairing-based homomorphic certificateless signature (CLS) technique is proposed by Yang et al. [15]. Bian et al. propose a privacy-preserving federated learning system that includes homomorphic encryption with a supplementary trust chain [16]. Zhou et al. [17] describe a differentially private federated learning approach against poisoning attacks, suitable for computing at the edge deployment.

A strong method that uses homomorphic encryption (HE) to guarantee data security during decentralized training is presented by Rahbari et al. [18]. Wanyu et al. propose Solitude, a novel privacy-preserving system that is based on graph neural networks (GNNs) with formal privacy guarantees based on edge local differential privacy [19]. To ensure that the on-chain privacy information is both readable and untraceable, Keke et al. [20] introduce a data privacy-saving scheme of cross-edge blockchain networks. That encrypts the data on the chain using completely homomorphic cryptography. In order to provide a cheap computation, a utility-privacy balanced approach, Harine et al. [21] combined a genetic algorithm using a deep learning network constructed using adversarial training. A lightweight edge computing data privacy-preserving approach based on blockchain and homomorphic encryption is proposed by Ling et al. [22]. Yuris et al.'s unique privacy-preserving FL approach [23] makes use of mobile edge nodes (MENs) and homomorphic encryption (HE) based approaches, especially resource-constrained MUs.

2.1. Limitations of Existing Approaches

While several privacy-preserving approaches for edge computing have been proposed in the literature, there are some challenges and drawbacks to the current study. Most of the current privacy-preserving disclosures, including differential privacy and homomorphic encryption, entail huge computation and communication overhead, and thus are not suitable for real-time applications on constrained edge points. However, most of the current solutions are oriented to secure data processing in transmission and storage, not secure computation, which is a challenge in edge computing. The incorporation of these techniques involves different trade-offs of privacy and efficiency, where enhanced privacy levels come hand-in-hand with computation time and resource consumption. For instance, though homomorphic encryption ensures high security of data, it poses severe computational pressure, reducing its implementations in actual edge computing scenarios. Similarly, next in MPC expect a great deal of device interaction, which presents challenges in edge networks that may afford limited bandwidth and high latency.

3. Proposed Architecture for Privacy-Aware Edge Computing

3.1. Overview of the Proposed Architecture

Here, we propose a privacy-oriented and performance-optimized edge computing model that computes sensitive data on the network edge and, in the meantime, ensures a privacy guarantee. The proposed architecture is explicitly structured to ensure that it reduces cryptographic overhead by selective encryption and maximizes privacy protection by adaptive privacy mechanisms, unlike traditional edge architecture schemes, which encrypt or protect privacy by relying on cloud-assisted privacy enforcement. This architecture allows a useful trade-off between computational performance and privacy protection, which makes it applicable to devices with resource constraints on edges that can be used to support real-time applications, including IoT networks, health monitoring, and processing financial data. The overall system workflow diagram is shown in Fig. 1. The architecture is built on a three-layered hierarchical model, with each layer having its own defined responsibilities in line with the contributions suggested:

Edge Device Layer: This is the layer that is composed of data-generating devices like IoT sensors, wearable devices, and smartphones. Unlike the conventional edge systems, edge devices in the proposed framework can do feature-level privacy classification during data acquisition. Lightweight homomorphic encryption is used to encrypt sensitive attributes and run non-sensitive attributes locally in plaintext. This selective encryption scheme is very beneficial as it minimizes computation and memory overheads and provides privacy enforcement at an early stage. Also, edge devices do local differential privacy noise injection with a dynamically calibrated noise scale, meaning that privacy protection is enforced before any data transmission.

Edge Server Layer: Edge servers handle calculation and aggregation between two or more edge devices. Instead of using raw or partially decrypted data, the edge servers do privacy-preserving aggregation of encrypted and noise-added intermediate values. This layer deploys a lightweight aggregation model, which implements real-time analytics without incurring the high communication and computation costs of secure multi-party computation. Notably, edge servers need not have access to plaintext data, and thus, end-to-end edge-level privacy enforcement is ensured.

Cloud Layer: The cloud layer is utilized only in long-term storage and massive analytics of privacy-preserved aggregated outcomes. The cloud in the proposed framework never takes in raw or sensitive data, including encrypted data, as it does in the traditional cloud-centric architecture. This design does not require a trusted cloud and is in compliance with the strict privacy rules like GDPR and HIPAA since it implements edge-only privacy protection.

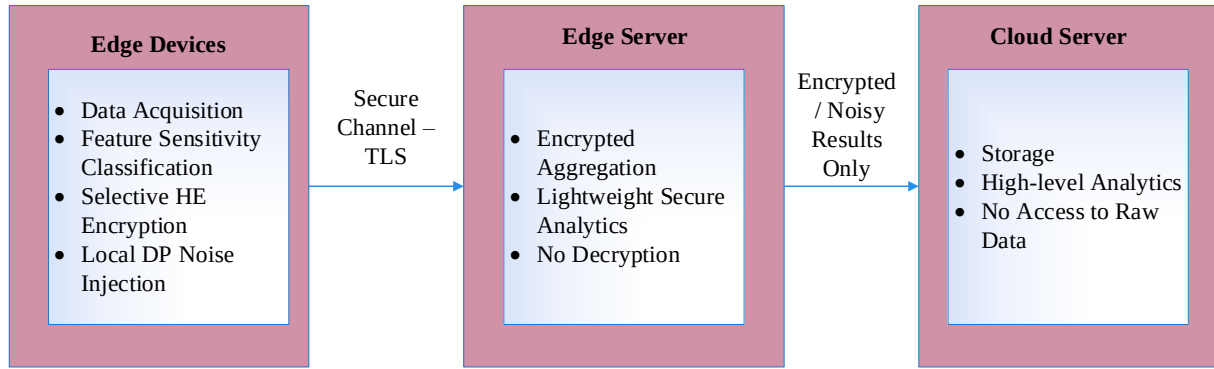


Fig. 1. Overall System Workflow Diagram.

3.2. System Components

The suggested architecture combines multiple closely coupled elements that jointly allow facilitating privacy-aware edge computing in an efficient and scalable way:

Selective Data Encryption Module: This module is an edge device-level module that features selective encryption. Lightweight homomorphic encryption applies only to encrypt privacy-sensitive data attributes, and non-sensitive data are not encrypted, enabling them to be processed locally. This method also has a great deal of cryptographic overhead reduction over full-data encryption models.

Privacy-Aware Computation Module: This module is implemented both on edge devices and edge servers and is used to compute encrypted data with lightweight homomorphic functions. It allows local computation and aggregation to be done with security and without exposing sensitive data, and real-time analytics with low latency.

Adaptive Differential Privacy Module: This module adapts the scale of the differential privacy noise in real-time depending on the contextual factors, including edge device count, aggregation frequency, and latency limits. The system is more of a middle ground between privacy protection and result accuracy as it dynamically varies the privacy strength.

Edge-Only Privacy Policy Manager: The privacy policy manager is an enforcement of the data access control, consent management, and regulatory compliance on the edge. The architecture removes the necessity to trust the providers of the cloud and increases adherence to regulations by ensuring that all decisions and protection related to privacy are made locally.

Secure Communication Channel: This ensures that all inter-layer communication is safe using secure protocols such as TLS. However, since the encryption of data and protection against privacy are already presupposed before transmission, the channel of communication is an additional protective measure, and not the primary privacy control.

Security model and threat assumptions: The system accepts that there is an honest-but-curious adversarial model, where the edge server can be interested in learning sensitive information out of processed data but obeys the given protocol. Homomorphic encryption is used to ensure data confidentiality, and the potential privacy leakage of the outputs is prevented by using the ADP. It is accepted that the private keys are kept safe and are inaccessible to the adversaries, and communication channels are secured against outside assaults.

3.3. Process of the Proposed Privacy-Aware Edge System

The proposed system workflow is developed to ensure end-to-end privacy protection at the edge, as well as real-time computation and scalable analytics. The proposed architecture is unlike the traditional edge cloud workflow, which sends raw or partially protected data to centralized servers, and where sensitive data is never leaked out of the edge layer, even during computation. The edge devices first of all gather raw data and do feature-level privacy classification, distinguishing between the sensitive and non-sensitive features. Sensitive features are encrypted instantly with selective lightweight homomorphic encryption (LHE), and non-sensitive features are kept in plaintext to decrease computation costs. Homomorphic operations are run on the encrypted sensitive features, and the local preprocessing and partial computations are performed without the public disclosure of the raw private data. Then, the encrypted intermediate results are sent to the edge server via secure communication channels (e.g., TLS). Privacy-sensitive computations and aggregations are done at the edge server on encrypted data without decryption. At this phase, context-aware differentiating privacy mechanism dynamically adds noise to aggregated results depending on current system circumstances like the number of devices involved, frequency of aggregation, and latency limitation. This avoids the inference attacks without compromising the output utility.

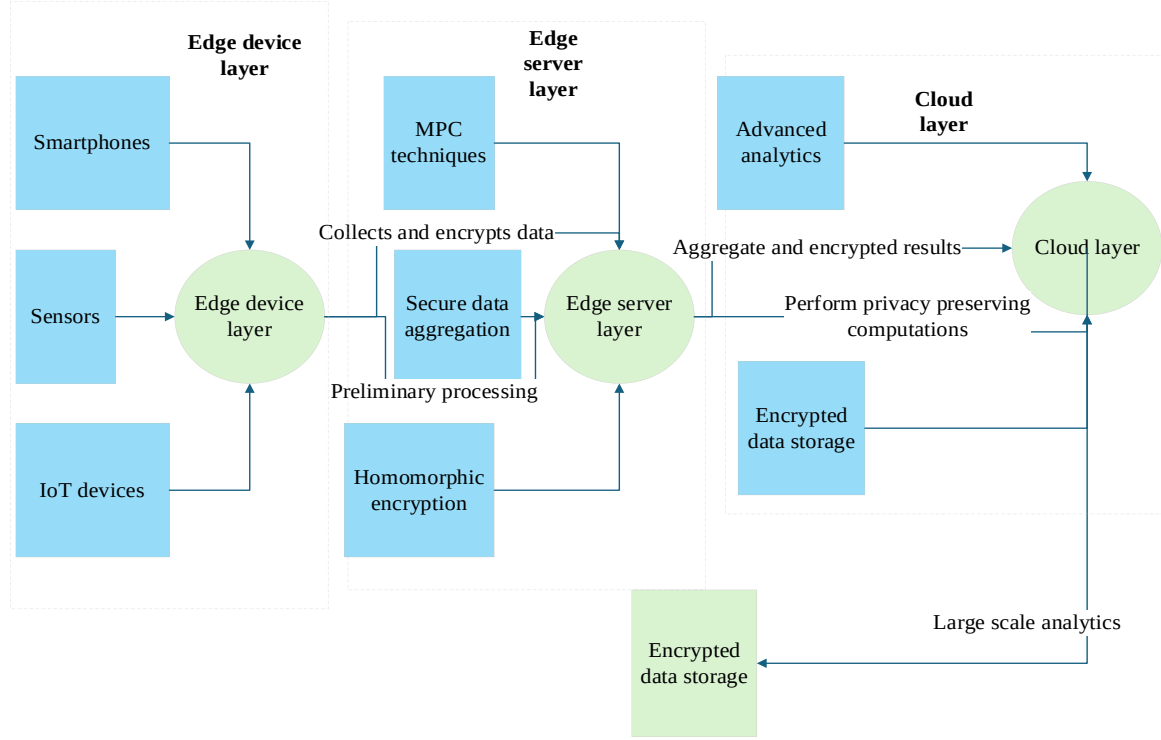


Fig. 2. Architecture for Privacy-Aware Edge Computing.

Fig. 2 presents the structure of the privacy-conscious edge computing system, which demonstrates how data moves through each layer and what privacy-preserving mechanisms are applied in each layer.

Selective Lightweight Homomorphic Computation. The majority of privacy-preserving edge computing systems are based on homomorphic encryption of the full dataset, which is a computationally expensive process and cannot be used in real-time by resource-limited edge computers. The proposed system to reduce this limitation is the introduction of a selective, lightweight homomorphic computation strategy [24]. In this method, data received in edge devices is initially analyzed in terms of sensitivity. Lightweight homomorphic encryption is only used to encrypt privacy-sensitive attributes, including medical measurements, financial indicators, or personally identifiable information, and process non-sensitive attributes in plaintext. This selective encryption also greatly minimizes cryptographic functions, memory, and processing latency. The proposed approach ensures high levels of confidentiality as only sensitive features are subject to homomorphic computation, and it is practical in the low-power edge settings.

Let the raw data collected by an edge device be represented as Eq. (1)

$$E = \{e_1, e_2, e_3, \dots, e_m\} \quad (1)$$

where partitioning the feature set is denoted as E , $e_1, e_2, \dots$ is represented as data attributes present in the feature sets and e_m is represented as the total number of attributes present in the feature set. into Sensitive and non-sensitive features. Sensitive features are denoted as $E_s \subset E$, and non-sensitive features are denoted as $E_{ns} \subset E / E_s$.

Only sensitive features are encrypted using a lightweight homomorphic encryption scheme using Eq. (2).

$$c_s = enc_{lhc}(E_s) \quad (2)$$

where, C_s is denoted as ciphertext sensitive features and enc_{lhc} represents an encryption function with respect to the lightweight homomorphic encryption. Non-sensitive features remain unencrypted: $E_{ns}^{pln} = E_{ns}$. Homomorphic computation is applied only to encrypted sensitive features in Eq. (3).

$$enc(x) \oplus enc(y) = enc(x + y) \quad (3)$$

where, $enc()$ is signified as a Homomorphic encryption function, $enc(x), enc(y)$ is denoted by ciphertext values. In addition Homomorphic multiplication is performed using Eq. (4).

$$enc(x) \otimes enc(y) = enc(x.y) \quad (4)$$

This makes the approach practical for low-power edge devices while maintaining strong confidentiality guarantees. The CKKS scheme is used to implement the SHE mechanism in this model; also, it allows approximate real arithmetic operations. The scheme supports both the additive and multiplicative homomorphic computations, and thus this model can be used in processing the data that is defined by edges. Addition is computationally efficient, but controlled depth enables multiplication to greatly reduce the amount of noise.

Dynamic Differential Privacy Noise Calibration. Conventional methods of differential privacy generally enforce a constant noise amount irrespective of the circumstances within the system, and as such, may lead to unnecessary loss of accuracy or lack of privacy. Conversely, the suggested system proposes a dynamic and context-sensitive differentiating privacy mechanism [25]. The added noise to the computation results is dynamically adjusted according to the real-time considerations of the number of participating edge devices, the rate of data aggregation, and the latency requirements of the application. In scenarios where the aggregation is of many devices or with repetitive queries, more privacy protection is ensured by adding more noise. In time-sensitive or low-risk situations, noise is minimized so that the result accuracy is maintained. This adaptive noise calibration allows achieving a better privacy-utility trade-off and enables the system to have high-quality analytics and strong privacy protection on the edge.

Let the aggregation function $f(E)$. In Eq. (5), the DP mechanism is assured.

$$pr[f(E) \in s] \leq e^\epsilon pr[f(E') \in s] \quad (5)$$

Where, $pr(.)$ is denoted as a probability function for any original dataset E , neighbouring datasets E' , S is possible outputs and e^ϵ privacy budget at the privacy level τ .

Adaptive Privacy Budget. The privacy budget is dynamically computed as Eq. (6).

$$\epsilon_t = \chi \cdot \frac{1}{n(t)} \cdot \delta \frac{1}{f(t)} + \alpha \cdot l(t) \quad (6)$$

Where, $n(t)$ number of participating edge devices, $f(t)$ aggregation frequency, $l(t)$ latency constraint, χ, δ, α system-defined weighting parameters.

Noise Injection: For numerical aggregation, Laplace noise is applied using Eq. (7).

$$\bar{f}(E) = f(E) + lap\left(\frac{\Delta f}{\epsilon_t}\right) \quad (7)$$

Where Δf is the global sensitivity of the function f , $\bar{f}(E)$ is noisy outputs for privacy maintaining, $f(E)$ is denotes true aggregation outputs and ϵ_t is a privacy budget parameter.

Edge-Only Privacy Enforcement Architecture. Compared to the current hybrid edge-to-cloud systems, which are based on the trusted cloud servers to aggregate or enforce privacy, the proposed architecture guarantees that all privacy-preserving operations are enforced at the edge [26]. Data encryption, privacy noise injection, and aggregation are either done on edge devices or edge servers, meaning that sensitive data is never sent off of the edge layer in either raw or partially processed form. There is a limit to the large-scale analytics or storage, where the cloud layer is limited to privacy-preserved and aggregated results only. The merit of such a privacy enforcement model is that it does not require any form of trust in cloud service providers, reduces the risk of centralized data breach, and adheres to inflexible data protection regulations such as GDPR and HIPAA. The architecture has the effect of bringing privacy management close to data sources, which provides more guarantees for sensitive applications.

Privacy Enforcement Flow. The edge device local encryption is determined by calculating by use of Eq. (8).

$$c_s = enc_{lne}(E_s) \quad (8)$$

Addition of edge-side noise is carried out by means of Eq. (9).

$$c_s^{dp} = c_s + \text{noise}(\varepsilon_t) \quad (9)$$

The secure aggregation $\mathbf{a}$ at edge servers is measured with the help of Eq. (10).

$$\mathbf{a} = \sum_{i=1}^n c_{s,i}^{dp} \quad (10)$$

At no stage does the cloud receive: E_s , C_s or individual updates. The CKKS scheme uses more ciphertext space than plaintext data because of the encoding of polynomials; however, the selective encryption approach lowers the total communication and storage costs as only sensitive features are encrypted. The edge device is issued with a pair of public-private keys, where the public key is utilized in encryption and computation, and the private key is safely stored and reserved for decryption. Key distribution is based upon a secure channel assumption. In a bid to control noise expansion in the homomorphic multiplication process, relinearization is used after multiplication operations, which guarantees correctness in the computation as well as acceptable noise levels.

Lightweight Privacy-Preserving Aggregation for Real-Time Edge Analytics. To support collaborative analytics and learning, secure aggregation over several edge devices is required, but the current solutions using fully homomorphic encryption or secure multi-party computation are computationally or communicationally expensive. The suggested system presents a privacy-sensitive aggregation model that is lightweight and combines adaptive differential privacy with selective homomorphic computation. All edge devices independently compute encrypted and privacy-resistant summaries, which are efficiently aggregated at the edge server without any individual contribution being disclosed [27]. The design allows scalable and real-time analytics of large volumes of edge devices with a high level of privacy assurance. The system can achieve high-throughput and low-latency edge analytics, which is impractical in current heavy cryptographic methods, by reducing the scope of encryption and dynamically managing privacy noise.

For n edge devices, each produces a perturbed encrypted update k_i in Eq. (11).

$$k_i = \text{enc}_{l_{he}} \left(f(E_{s,i}) \right) + dp_i \quad (11)$$

Where, dp_i is denotes various privacy concerns at edge devices i . The edge server computes () using Eq. (12).

$$K = \sum_{i=1}^n k_i \quad (12)$$

The homomorphic properties are performed using Eq. (13).

$$K = \text{enc}_{l_{he}} \left(\sum_{i=1}^n f(E_{s,i}) \right) + \sum_{i=1}^n dp_i \quad (13)$$

These contributions combined form a feasible, secure edge computing model that breaks several major constraints of current methods. The suggested system will lessen the computational load, adjust privacy protection to real-time conditions, avoid the use of trusted cloud elements, and allow real-time analytics to scale. The features of the framework render it especially appropriate to the latency-sensitive applications, including smart healthcare, financial monitoring, and large-scale IoT systems.

Start

{

Phase 1: Data Acquisition and Feature Classification (Edge Device)

for each edge device i do

{

Acquire raw data d_i

Identify sensitive features $S_i \subseteq d_i$ based on the privacy policy p

```

    Identify non-sensitive features  $n_i = d_i - s_i$ 
    Encrypt  $s_i$  using Lightweight Homomorphic Encryption  $\rightarrow enc(s_i)$ 
    Keep  $n_i$  in plaintext for local processing.
}
End for
Phase 2: Local Privacy-Preserving Computation (Edge Device)
  for each edge device  $i$  do
  {
    Perform homomorphic operations on  $enc(s_i) \rightarrow enc(\hat{c}_i.s_i)$ 
    Perform normal computation on  $n_i \rightarrow \hat{c}_i.n$ 
    Combine results
  }
  End for
Phase 3: Adaptive Differential Privacy Noise Injection (Edge Device)
  for each edge device  $i$  do
  {
    Estimate system context  $C$  (number of devices, latency, risk level)
    Dynamically select privacy budget  $\mathcal{E}_i \leq \mathcal{E}_{\max}$ 
    Generate DP noise  $noise_i$  based on  $\mathcal{E}_i$ 
    Apply noise to the computation result.
  }
  End for
Phase 4: Secure Aggregation (Edge Server)
  Initialize the aggregated result  $agg = 0$ 
  for each received  $C'_i$  from edge devices do
  {
    Aggregate encrypted and noisy results  $\rightarrow agg = agg \oplus C'_i$ 
  }
  End for
Phase 5: Cloud-Level Analytics (Cloud Layer)
  {
    Transmit the aggregated result  $Agg$  to the cloud.
    Perform high-level analytics on  $Agg$ 
    Store aggregated privacy-preserved results.
    Return final output  $r\_cloud$ 
  }
}
End

```

3.4. Privacy and Security Mechanisms

The proposed architecture combines a collection of sophisticated and dynamic privacy and security strategies to provide high privacy protection as well as computational efficiency with resource-constrained edge environments. Contrary to traditional edge computing systems that are based on either static encryption or centralized privacy enforcement, the system proposed implements context-aware, edge-based privacy preservation across the data lifecycle.

Selective Lightweight Homomorphic Encryption: The architecture proposed uses a selective lightweight homomorphic encryption (LHE) scheme, in which only privacy-sensitive data attributes are encrypted, and not the whole dataset. This allows one to compute securely on encrypted data and also aids in minimizing the cryptographic overhead to a great extent. The process of selective encryption is done depending on the analysis of data sensitivity. The

predefined thresholds are used to categorize features in terms of sensitivity or non-sensitivity rules. Sensitive data is encrypted with SHE, whereas data that is not sensitive is processed in plain text to reduce computational costs. This is the selection mechanism that is executed at the data pre-processing stage.

Context-Aware Differential Privacy: To avoid inference attacks of aggregated outputs, the architecture includes a dynamic differential privacy mechanism. In contrast to the classical methods that use a predetermined privacy budget, the suggested system can increase and decrease the scale of noise depending on the sensitivity of the data, the frequency of aggregation, and the number of involved edge devices.

Federated Learning with Edge-Only Secure Aggregation: To combine both federated learning and analytics, the presented system incorporates federated learning and edge-only secure aggregation. The model updates are computed in the edge devices on selectively encrypted data and perturbed with adaptive differential privacy, and then aggregated. The aggregation is done at the edge server, which lacks access to the individual updates or raw data, and the edge server or the cloud cannot deduce sensitive information. The step-by-step process of the data processing workflow is shown in fig.3.

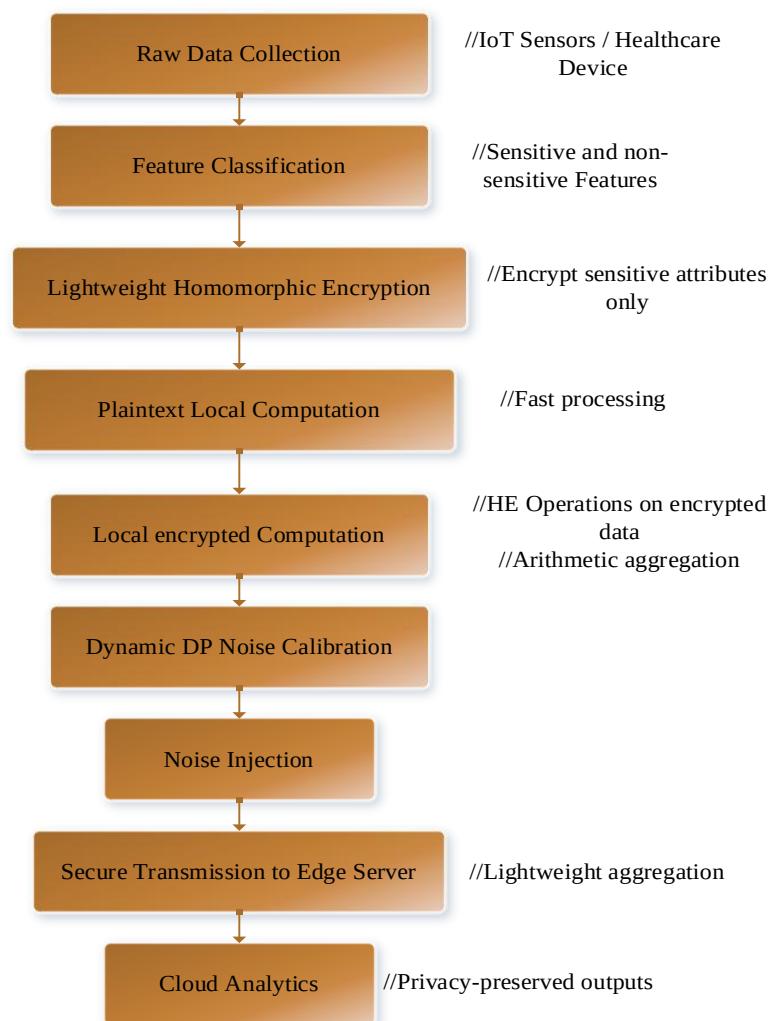


Fig. 3. Step-by-Step Data Processing Workflow.

The proposed model includes an adaptive differentiating privacy mechanism whereby the privacy budget parameters are spelled out to regulate the degree of privacy protection. Privacy budget is used to decide the level of noise to be introduced into the output, hence a balance between privacy protection and data utility is achieved. The proposed method dynamically changes the privacy level depending on the conditions of the sensitivity of the data and the workload, in contrast to conventional methods that rely on a fixed privacy budget. In particular, privacy protection is further enhanced by making noise levels higher on highly sensitive data or critical operations, whereas less sensitive data is handled with less noise to ensure the output is accurate and useful. The dynamic approach to privacy can effectively control privacy on real-time edges. The suggested adaptive process has guaranteed differential privacy where the noise is applied consistently and in a controlled way in all its operations. The total loss of privacy is well

controlled to operate within the acceptable limits, even when there is more than one query or running data processing. This would guarantee that the output does not imply any sensitive information, thus offering great privacy protection.

3.5. Advantages and novelty of the Proposed Architecture

The suggested privacy-conscious edge computing system is a new, effective approach that combines feature-level selective homomorphic encryption, context-specific differentiated privacy, and edge-wide privacy control into a single system. In contrast to the current methods that would encrypt whole datasets of data with a unified encryption scheme or a fixed privacy control mechanism, the proposed model encrypts only sensitive attributes with a lightweight homomorphic encryption algorithm and lets non-sensitive data be processed in plaintext at the edge to greatly decrease the amount of computational power and latency. Besides, the framework proposes a dynamic differentiable privacy mechanism that dynamically scales the noise according to the real-time context of the system, which provides a better privacy-accuracy trade-off without affecting the responsiveness. Another crucial novelty is implementing all privacy-saving operations solely on the edge without any dependency on trusted cloud computation and making sure that no information that is sensitive at all leaves the edge layer in any form. Collectively, these innovations allow scalable, real-time, and regulation-compliant analytics of resource-constrained edge devices and make the suggested framework both pragmatically efficient and technically different than the current privacy-preserving edge and cloud computing systems. The novelty of the proposed work is the context-aware combination of SHE and ADP in an edge computing environment. In contrast to existing hybrid approaches to HE-DP, which are based on full encryption and a fixed privacy budget, the suggested one includes selective encryption due to the sensitivity of the data and the reduced computational cost, and an adaptive privacy strategy that dynamically responds to data properties and conditions of the system. This is a coordinated design, which allows the framework to effectively trade off privacy preservation, computational efficiency, and latency, which is especially appropriate in real-time and resource-constrained edge scenarios.

4. Experimental Setup

In this section, the experimental design of the proposed privacy-aware edge computing architecture and the algorithm are described to measure the performance and efficiency of both. The experiments will be used to measure different parameters like encryption time, computational efficiency, communication overhead, and privacy assurances that the algorithm offers. The parameters utilized in the experimental setup are summarized in the following table 1.

Table 1. Parameters and values used in the experimental setup.

Parameter	Value
Processor	Intel core i7
Platform	Python
Operating system	Windows
RAM	16GB
Dataset type	Healthcare
Edge Device Count	50 (varied between 10 and 500 for scalability)
Dataset Size	10,000 records (healthcare data)
Computation Type	Basic arithmetic operations (summation, etc.)
Noise Scale for DP	Laplacian noise with scale 1.0
Encryption Type	Lightweight Homomorphic Encryption
Communication Protocol	TLS for secure communication
Evaluation Duration	24 hours per dataset
Type of task	Secure computation with encrypted healthcare data
Computational overhead	Based on the processing cost
Latency	Total time required for each computation
Hybrid HF-DP	Fixed privacy settings and encryption

4.1. Dataset Description

In the experiments, we have a simulated dataset of sensitive data that we usually perform on the edge computing environment. The data is structured in a way that it resembles real-world applications, including healthcare monitoring, financial transactions, and IoT sensor data. The following are the specifications of the dataset used to conduct this experiment:

- **Dataset 1: Synthetic Healthcare Dataset [28]**- Dataset 1 contains 10000 records, which include various patient demographics and medical conditions, and potential vital signs as their patient-related attributes. The

educational and research purposes of this dataset make it suitable for the edge-computing experiments because it functions as a simulation tool.

- **Dataset 2: Bangladesh IoT Environmental Dataset (BIED) [29]**– Dataset 2 includes continuous IoT sensor data, which monitors environmental conditions such as temperature and humidity, air quality index AQI and CO₂ levels, soil moisture, and additional factors. The system provides authentic time-series IoT data, which researchers can utilize to test and analyze large sensor networks through their modeling process.

Both datasets are sensitive, and high privacy assurance in the processing of the data is necessary, which will be suitable for testing the privacy-conscious capability of the proposed architecture.

4.2. Experimental Environment

The experiments are run on a simulated edge computing environment that has edge devices and edge servers. The hardware/software configurations are as follows in table 2.

Table 2. Hardware/Software configurations.

Layer	Component	Specification
Edge Device Simulation	Number of Devices	50 edge devices
	CPU	ARM Cortex-A53, quad-core, 1.4 GHz
	RAM	1 GB
	Operating System	Raspberry Pi OS
	Software Environment	Python 3.9
Edge Server Simulation	CPU	Intel Xeon E5-2650, 10-core, 2.30 GHz
	RAM	32 GB
	Operating System	Ubuntu 20.04
	Software Environment	Python 3.9
Cloud Server	CPU	Intel Xeon E5-2670, 16-core, 2.60 GHz
	RAM	64 GB
	Operating System	Ubuntu 20.04
	Software Environment	Python 3.9

4.3. Evaluation Metrics

The proposed architecture and algorithm have their simulation and empirical results evaluated based on several parameters. The Encryption Time is the time that the individual edge devices require to encrypt the sensitive data before computation, and it is useful in the strategy of improving low latency in applications for real-time. The latter, computation time, evaluates the real-time computational performance of local computations on encrypted data in the edge devices. The time spent on adding differential privacy noise to the results provides privacy, but at the cost of computation efficiency. Communication latency is the time to transmit encrypted data between edge devices, edge servers, or the cloud, and it should be minimal due to the requirement of processing data fast in an edge environment. Resource Utilization quantifies CPU and memory utilization during the encryption, computation, and generation of noise to understand how real devices with limited computational power are to be operated. The Accuracy vs Privacy Trade-off assesses the impact of introducing differential privacy noise to the process of aggregation and sets privacy vs result accuracy. Finally, Scalability examines how the system uses additional edge devices and how it copes with the increased size (10 to 500 devices) of the system.

4.3.1. Encryption Time Analysis

The encryption time is a business-level parameter that is a great concern to real-time edge computing applications because cryptographic overhead can drastically reduce the responsiveness of the system. Lightweight homomorphic encryption is applied selectively in the proposed architecture to privacy-sensitive features, and this is one of the direct reasons that result in a lower cost of encryption.

Fig. 4 shows the average time to encrypt each record by varying datasets and numbers of edge devices. True to expectations, the encryption time is also dependent on the size of the dataset, as well as the number of devices involved, due to the larger amount of data being processed at once. Nevertheless, in the most challenging case of 10,000 records and 500 edge devices, the encryption time is less than 30 ms, which is far less than the acceptable range of real-time edge applications.

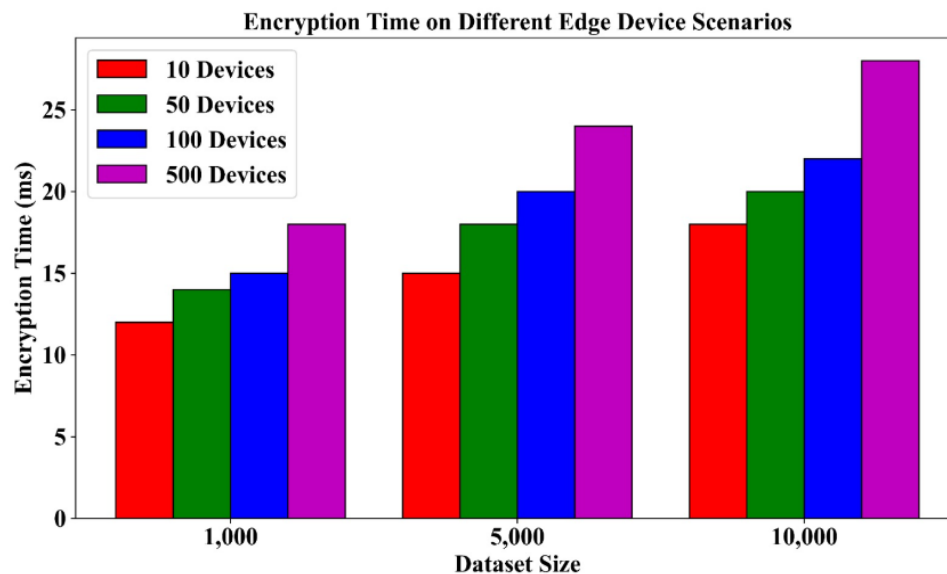


Fig. 4. Encryption time analysis with different edge device scenarios.

4.3.2. Computation Time Analysis

Computation time is the time that is needed to carry out basic arithmetic tasks (like addition and summation) on encrypted data at the edge devices. This metric is the measure of the practicality of conducting privacy-preserving analytics in real time.

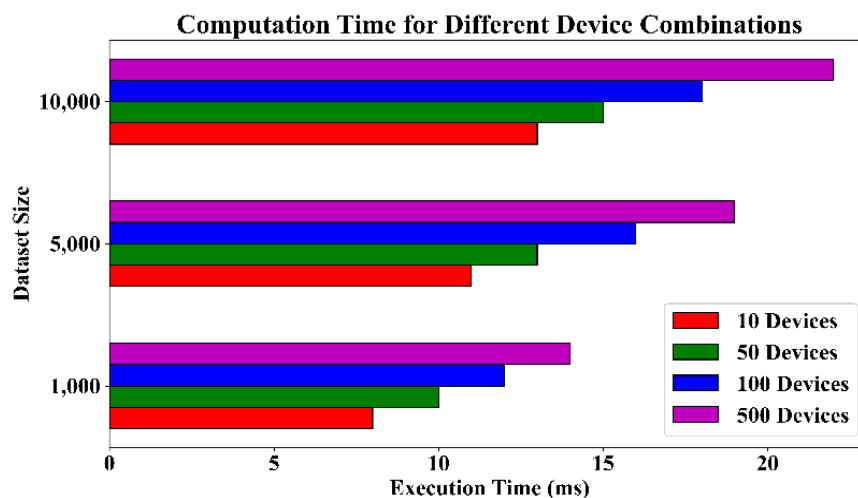


Fig. 5. Computation time analysis with different edge device scenarios.

The increase in computation time with dataset size and the number of edge devices, as indicated in Fig. 5, as expected, is in correspondence with the higher computational workload. However, the total computation time is very low at a maximum of 22 ms with 10,000 records and 500 devices.

4.3.3. Differential Privacy Noise Addition Overhead

The addition of noise during the process of data aggregation is a crucial aspect of preventing the occurrence of inference attacks. Nevertheless, too much noise calculation can add undesirable latency. Fig. 6 shows that additional time to inject DP noise at other settings is possible.

The findings indicate that the overhead on noise addition is small (between 2 ms and 8 ms), though the dataset size and the number of devices increase. This overhead is a very small fraction of the total computation time, and it implies that the dynamic noise calibration mechanism is computationally efficient.

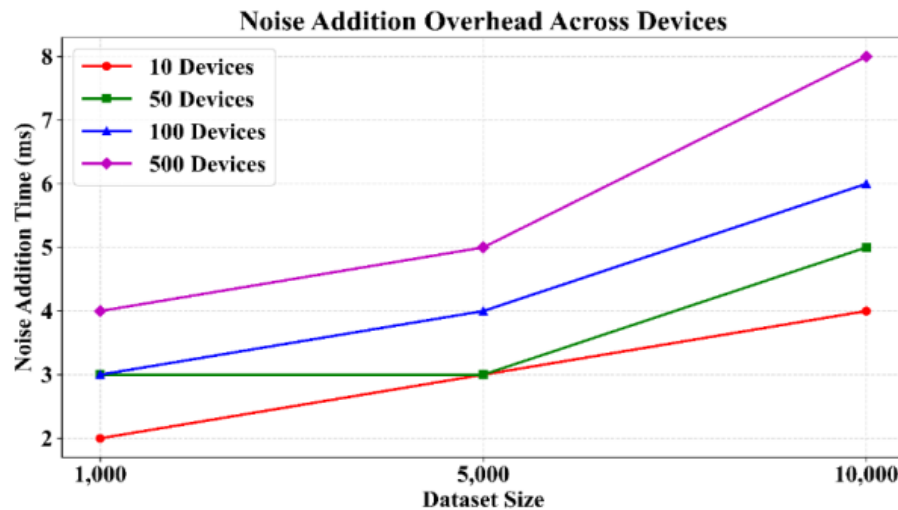


Fig. 6. Analysis of Noise Addition Overhead.

4.3.4. Communication Latency Evaluation

Communication latency is a metric that is used to estimate the duration of transmitting encrypted and noisy results between edge devices, edge servers, and the cloud. This measure is especially significant in the comparison of edge-based solutions with conventional cloud-centric solutions.

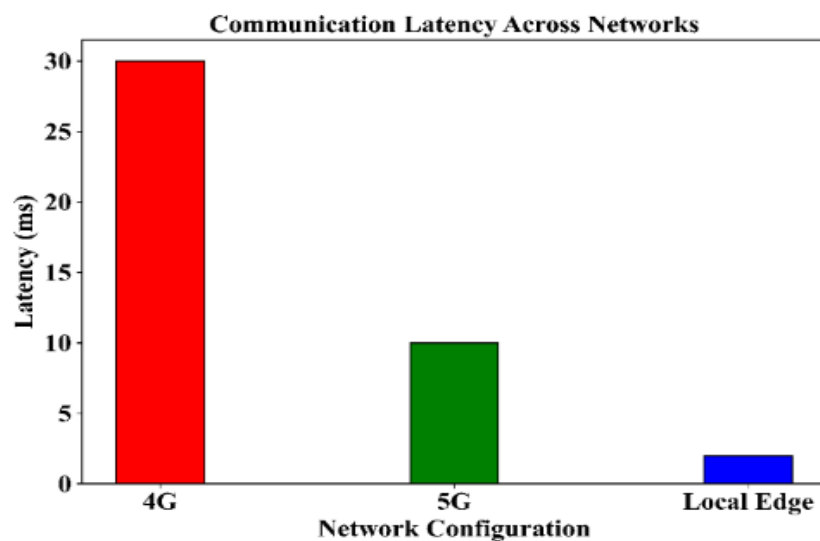


Fig. 7. Analysis of communication latency.

Fig. 7 demonstrates that the proposed system has the minimum possible latency when implemented in a local edge network with the average delay of 2 ms, as opposed to 10 ms in 5G networks and 30 ms in 4G environments. These results support the effectiveness of Localized edge processing that minimizes the utilization of long data transmission.

4.3.5. Resource Utilization Analysis

The one important metric that shows the feasibility of the implementation of the proposed framework on the limited edge devices is the resource usage. Table 3 shows a summary of CPU and memory consumption during encryption, local computation, and differential privacy noise addition.

Table 3. Resource Utilization.

Task	CPU Usage (%)	Memory Usage (MB)
Encryption	30	120
Local Computation	25	100
Noise Addition (DP)	10	50

The findings suggest that encryption uses the most resources of 30% CPU and 120 MB of memory, then it is succeeded by local computation and noise addition. Nevertheless, resource use in general is within the reach of common edge devices, including IoT sensors, single-board computers, and mobile devices.

4.3.6. Accuracy vs. Privacy Trade-off Analysis

The accuracy-privacy trade-off is considered through the quantification of the effect of the different levels of differential privacy noise on the accuracy of the aggregated outcomes. As can be seen in Table 4, the accuracy gradually decreases with the noise scale, which is a natural phenomenon in privacy-preserving systems.

Table 4. Accuracy vs. Privacy Trade-off.

Noise Scale	Accuracy (%)	Latency (ms)	Computation time (ms)
0.5	99.2	1.8	14
1.0	98.7	2.1	16
1.5	97.5	2.3	18
2.0	95.3	2.7	20

Nonetheless, the system has an accuracy of more than 95% at a relatively high noise scale of 2.0, which is satisfactory to a number of practical edge computing systems.

4.3.7. Aggregation Time

Fig. 8 quantitatively shows how the proposed architecture has a scalability advantage. With a participation of 10 edge devices, the aggregation time of the proposed system is 5 ms, which is less than 9 ms in the case of the existing edge solutions and 25 ms in the case of cloud-based processing. When the number of devices reaches 50, the time of aggregation is reduced by a small margin to 8 ms, compared to 15 ms with current edge solutions and 40 ms with cloud-based systems. The proposed architecture has a low aggregation time of 12 ms and 18 ms, even in large scale with 100 and 500 devices, respectively, compared to 22 ms and 40 ms with existing edge solutions and 65 ms and 120 ms with cloud-based processing. These findings validate the fact that local aggregation and edge-only privacy enforcement can be effective in minimizing the coordination and communication overhead.

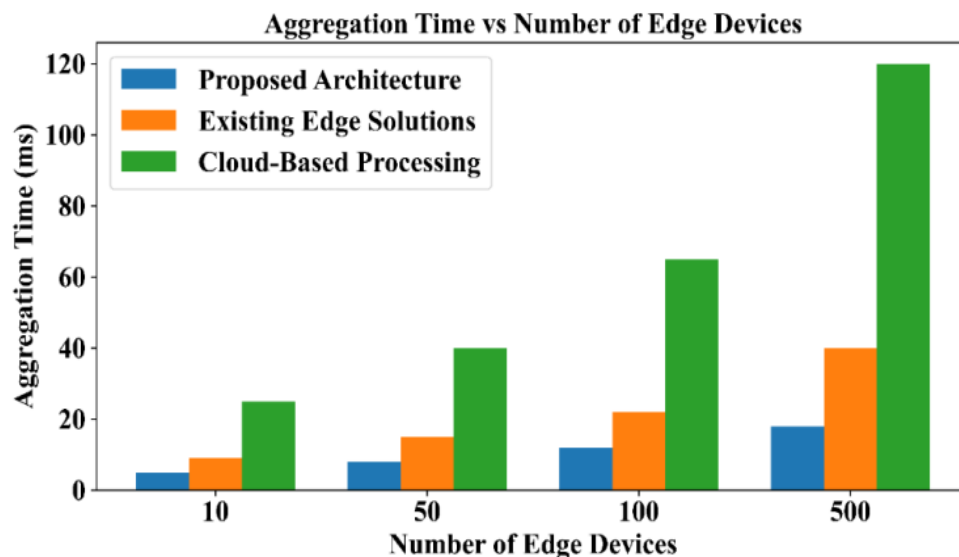


Fig. 8. Aggregation Time vs. Number of Edge Devices

4.3.8. Communication Cost per Aggregation Round

Fig. 9 reveals that cloud-based systems are characterized by a data transfer rate of about 120 MB/aggregation round and a latency of 50 ms, which implies high dependence on centralized processing. The current edge solutions decrease the data relayed to 60 MB and the latency to 10 ms. The suggested architecture also reduces the communication to a minimum, where only 20 MB of data is transferred per round of aggregation, and the latency is reduced to ultra-low, 2 ms. This is equivalent to an 83 percent decrease in the data transfer and a 96 percent decrease in the latency over cloud-based processing.

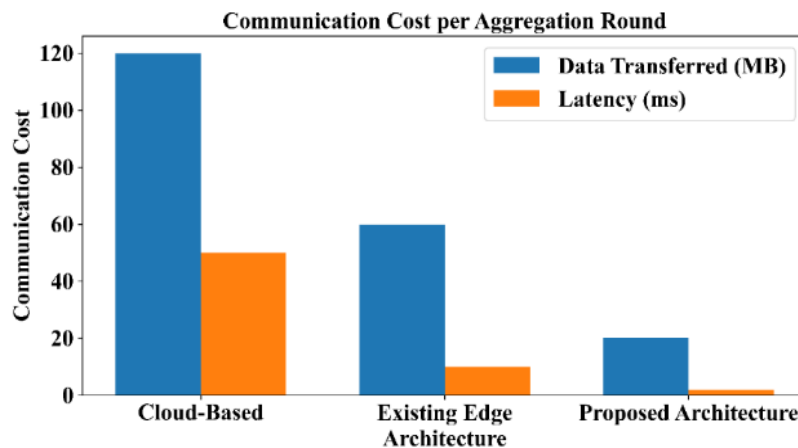


Fig. 9. Comparison of communication cost.

4.3.9. Encryption Overhead Comparison

Fig. 10 demonstrates the effectiveness of the suggested selective homomorphic encryption strategy. Full homomorphic encryption takes 35 ms to encrypt, 55-percent of CPU, and 240 MB of memory, which is prohibitive on edge devices. There is an existing edge encryption that decreases these values to 25 ms, 40% CPU, and 160 MB memory. The proposed selective HE has the lowest overhead and an encryption time of 18 ms, CPU usage of 30, and memory consumption of 120 MB. This is a 49 percent and 50 percent decrease in CPU and memory, respectively, over full homomorphic encryption and indicates its appropriateness to resource-constrained environments.

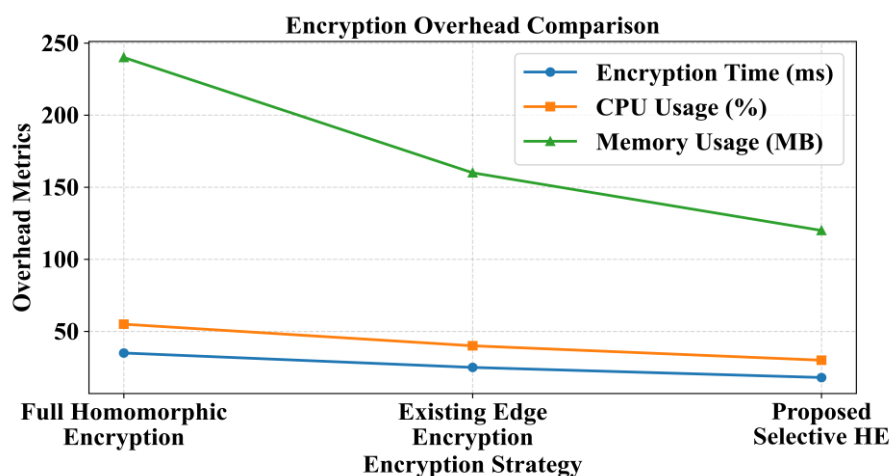


Fig. 10. Analysis of Encryption Overhead.

4.3.10. Energy Consumption per Task

Fig.11 shows energy usage between existing edge solutions and the proposed architecture. The energy cost per task is lower from 1.8 J to 1.1 J in the case of encryption, by 1.4 J to 0.9 J in the case of computation, and by 0.6 J to 0.4 J in the case of noise addition. The total energy per task is decreased by an amount of 37 percent. Such a significant power consumption reduction proves that the suggested architecture can be used in battery-powered and long-running edge deployments.

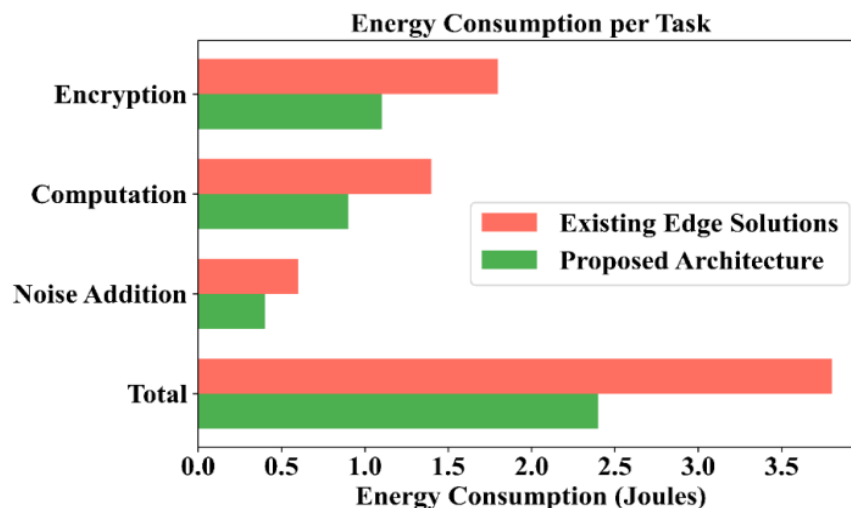


Fig. 11. Analysis of Energy Consumption.

4.4. Comparison with Baseline Systems

The comparison of the performance of the proposed architecture is made with two baseline systems: cloud-based conventional processing and other conventional Distillation with Selective Homomorphic Encryption (DSHE) and Multi-Agent Reinforcement Learning. Table 5 shows the outcomes of performance measures.

Table 5. Comparison with Baseline Systems.

Metric	Proposed Architecture	DSHE	MARL
Encryption Time (ms)	17 $\pm$ 1.21	21 $\pm$ 1.4	20 $\pm$ 1.32
Computation Time (ms)	14 $\pm$ 0.8	27 $\pm$ 2.2	18 $\pm$ 1.4
Communication Latency (ms)	21 $\pm$ 0.4	52 $\pm$ 3.4	10 $\pm$ 1.2
Privacy Guarantee	High (HE + DP)	Moderate (Encryption Only)	Moderate
Resource Utilization (CPU)	30% $\pm$ 2.1	50% $\pm$ 3.5	40% $\pm$ 2.5

Table 5 offers a comparative analysis of the suggested privacy-aware edge computing architecture and the traditional cloud-based processing and available edge computing applications in terms of the main performance and privacy indicators. These findings are clear evidence that the proposed architecture can achieve lower encryption time (18 ms) and computational time (15 ms) compared to both cloud-based solutions and current edge solutions, which validates the success of the selective lightweight homomorphic encryption and localized computation approach. Communication latency is one of the areas that can be especially improved, as the proposed architecture achieves only 2 ms compared to 50 ms in cloud-based processing and 10 ms in the current edge approaches, which illustrates the benefits of fully executing privacy-preserving operations on the edge. Regarding privacy protection, the proposed system offers a high privacy assurance level as it combines homomorphic encryption and differential privacy, whereas the baseline systems have a primary protection of inference attack only due to the use of encryption, which is only moderately protective. The suggested architecture also has a reduced CPU usage (30%), which means more resource efficiency and its applicability in deployment to a limited edge device. The proposed system has a competitive accuracy of 98.7 which is obtained, despite the stronger privacy guarantees, despite the fact that, although cloud-based processing has slightly higher accuracy, the proposed system is also competitive. Table 6 shows the analysis of security against the threat models.

Table 6. Analysis of security against the threat models.

Type of threat	Without protection	With proposed (SHE+ADP)
Side-channel leakage	120	35
Inference attacks	83%	52%
Collusion attack	0.56	0.06
Encryption security level	128bit	256bit
Honest curious nodes	0.64	0.04

On the whole, in this comparison, it is possible to state that the proposed architecture represents a good trade-off between latency, privacy, resource use, and accuracy, which makes it a more scalable, efficient, and reliable solution in privacy-aware edge computing in large-scale applications of the IoT and healthcare systems. Table 7 shows the analysis of encryption scope

Table 7. Analysis of encryption scope.

Encryption scope	Encrypted data (%)	Accuracy (%)	Latency (ms)	CPU usage
Minimum selection	20	88	1.8	60
Higher sensitivity	60	92	2.7	73
Full HE model	100	94	4.6	86

4.5. Discussion

Experiments were performed to validate the scalability of the proposed framework in edge environments that were resource-constrained, with respect to the size of the data as well as the level of concurrency. The findings indicate that the selective encryption approach is much lower-power-consuming than full homomorphic encryption, and it can work efficiently even with a large-scale IoT user base. The performance of the proposed framework has shown that its latency growth is lower than that of full homomorphic encryption as the amount of data grows. Furthermore, the system can be used in large-scale IoT implementation because of the selective encryption under high concurrency levels to ensure a stable performance of the system.

A healthcare IoT technology is developed to validate the proposed framework, which relies on the ongoing monitoring of the patient in terms of heart rate, blood pressure, temperature, and activity rates using wearable sensors. Selective encryption is used in this context on the basis of data sensitivity. Sensitive data, such as patient identity, heart rate, and medical history, which are encrypted with SHE, and non-sensitive data like ambient temperature, which is computed in plaintext to lower the amount of computation. In addition, application-specific rules are used to define sensitivity classification, with the attributes that exceed predefined privacy thresholds indicated as being encrypted. The homomorphic operations are applied to the encrypted data at the edge node, and ADP is employed to protect the results, and then they are given to outside systems. This function shows that the proposed model is suitable for finding a balance between privacy and the efficiency of computing in real-time healthcare settings.

5. Conclusion

This article introduced a new and effective privacy-aware edge computing system to solve the inherent trade-off between data confidentiality and system execution in resource-limited edge computing settings. The proposed architecture is able to cut down on cryptographic overhead by a significant margin by introducing feature-level selective lightweight homomorphic encryption, and therefore leaves sensitive data intact during the computation process. Moreover, the context-sensitive differential privacy framework is able to dynamically configure the strength of privacy depending on the conditions of the system, and thus provides a better trade-off between privacy assurances and the accuracy of the results. The main peculiar feature of the proposed framework is that it has a fully edge-enforced privacy model, which does not require trusted cloud computation and ensures that no sensitive data is transferred to the edge layer, which makes the security and regulatory compliance much stronger.

Extensive experimental analyses have shown that the proposed system has lower encryption and calculation time, low latency in communication, and better scalability than conventional cloud-based processing and other available edge computing solutions. Although the level of privacy is high, the framework is also characterized by high levels of analytical accuracy and minimal resource usage, which is why it applies to real-time settings, including healthcare monitoring, IoT analytics, and financial data processing. In general, the suggested architecture provides a viable and scalable framework for the next-generation privacy-preserving edge computing systems that provide high privacy protection without compromising performance. Future research will discuss the adaptive learning-driven privacy control and real-world implementation on the heterogeneous edge platforms.

All the Declarations and Statements

Author Contributions Statement

Apurva Khandekar – Writing – Drafted the initial manuscript, contributed to the literature survey, and documented the technical background of the study, Writing – Review and Editing, and Project Management: Reviewed and edited the manuscript, ensured clarity and coherence, and helped coordinate project milestones and deadlines.

Prathipati Ratna Kumar – Conceptualization, Methodology, and Supervision: Proposed research ideas, Constructed the overall framework, and supervised project execution, Data Curation and Software Implementation: Handled data

acquisition, dataset preprocessing, and implementing the research model, Model Training, Validation, and Performance Evaluation: Led the model training process, validated results using standard metrics, and benchmarked performance against existing methods, Formal Analysis, Visualization, and Statistical Analysis: Performed in-depth analysis of experimental results, prepared performance charts, and ensured the statistical robustness of the evaluation.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

No funding is provided for the preparation of manuscript.

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Ethical Declarations

This article does not contain any studies with human participants or animals performed by any of the authors.

Acknowledgments

We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

The authors declare that generative artificial intelligence (AI) tools were used only for language editing and improving the clarity of the manuscript. The AI tools did not contribute to the research design, data collection, data analysis, interpretation of results, or the development of the scientific content. The authors take full responsibility for the accuracy, originality, and integrity of the manuscript.

Notations

Notations	
E	partition the feature set
$e_1, e_2, \dots$	data attributes
e_m	total number of attributes
E_s	Sensitive features
E_{ns}	Non-Sensitive features
C_s	ciphertext sensitive features
enc_{he}	lightweight homomorphic encryption function
E_{ns}^{ph}	Unencrypted Non-Sensitive features
$enc()$	Homomorphic encryption function
$enc(x), enc(y)$	ciphertext values
$f(E)$	aggregation function
$pr(.)$	probability function
E	original dataset
E'	neighbouring datasets
S	possible outputs
e^τ	Privacy budget at the privacy level
$n(t)$	number of participating edge devices
$f(t)$	aggregation frequency
$l(t)$	latency constraint
χ, δ, α	system-defined weighting parameters
Δf	global sensitivity of the function

$f(E)$	true aggregation outputs
$\tilde{f}(E)$	noisy outputs
ϵ_t	privacy budget parameter
C_s^{db}	ciphertext sensitive features with DP mechanism
$\mathcal{a}$	Secure aggregation
dp_i	Various privacy concerns at edge devices i

Abbreviations

The following abbreviations are used in this manuscript:

IoT - Internet of Things
 HEC - Hybrid Edge Cloud
 ECC - Elliptic Curve Cryptography
 CLS - Certificateless Signature
 HE - Homomorphic Encryption
 GNNs - Graph Neural Networks
 LHE - Lightweight Homomorphic Encryption

References

- [1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019. doi: 10.1109/JPROC.2019.2918951.
- [2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016. doi: 10.1109/JIOT.2016.2579198.
- [3] T. A. Bablu and M. T. Rashid, "Edge computing and its impact on real-time data processing for IoT-driven applications," *J. Adv. Comput. Syst.*, vol. 5, no. 1, pp. 26–43, 2025.
- [4] M. Trigka and E. Dritsas, "Edge and cloud computing in smart cities," *Future Internet*, vol. 17, no. 3, p. 118, 2025.
- [5] A. Dalal, "Optimizing edge computing integration with cloud platforms to improve performance and reduce latency," SSRN 5268128, 2025.
- [6] C. Caiazza, S. Giordano, V. Luconi, and A. Vecchio, "Edge computing vs centralized cloud: Impact of communication latency on the energy consumption of LTE terminal nodes," *Comput. Commun.*, vol. 194, pp. 213–225, 2022.
- [7] Q. Xie, S. Jiang, L. Jiang, Y. Huang, Z. Zhao, S. Khan, and K. Wu, "Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey," *IEEE Internet Things J.*, vol. 11, no. 14, pp. 24569–24580, 2024.
- [8] A. Ali, B. A. S. Al-Rimy, F. S. Alsubaei, and A. A. Almazroi, "Healthlock: Blockchain-based privacy preservation using homomorphic encryption in IoT healthcare applications," *Sensors*, vol. 23, no. 15, p. 6762, 2023.
- [9] H. Kaminaga, F. M. Awaysheh, S. Alawadi, and L. Kamm, "MPCFL: Towards multi-party computation for secure federated learning aggregation," in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput.*, Dec. 2023, pp. 1–10.
- [10] T. A. Bablu and M. T. Rashid, "Edge computing and its impact on real-time data processing for IoT-driven applications," *J. Adv. Comput. Syst.*, vol. 5, no. 1, pp. 26–43, 2025.
- [11] M. Boopathi, S. Gupta, A. M. Zabeculla, R. Gupta, V. Vekriya, and A. K. Pandey, "Optimization algorithms in security and privacy-preserving data disturbance for collaborative edge computing social IoT deep learning architectures," *Soft Comput.*, pp. 1–13, 2023.
- [12] F. Tusa, "Privacy-preserving FaaS: A marketplace for the serverless edge-cloud continuum," in *Proc. 34th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2025, pp. 1–2.
- [13] S. M. Alamouti, F. Arjomandi, and M. Burger, "Hybrid edge cloud: A pragmatic approach for decentralized cloud computing," *IEEE Commun. Mag.*, vol. 60, no. 9, pp. 16–29, 2022.
- [14] A. Alzahrani, "Developing a provable secure and cloud-centric authentication protocol for the e-healthcare system," *IEEE Access*, 2024.
- [15] F. Xu, S. Liu, and X. Yang, "An efficient privacy-preserving authentication scheme with enhanced security for IoMT applications," *Comput. Commun.*, vol. 208, pp. 171–178, 2023.
- [16] B. Zhu and L. Niu, "A privacy-preserving federated learning scheme with homomorphic encryption and edge computing," *Alexandria Eng. J.*, vol. 118, pp. 11–20, 2025.
- [17] J. Zhou, N. Wu, Y. Wang, S. Gu, Z. Cao, X. Dong, and K. K. R. Choo, "A differentially private federated learning model against poisoning attacks in edge computing," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 1941–1958, 2022.
- [18] D. Rahbari, M. Daneshmand, and M. Jenihhin, "An efficient architecture for edge AI federated learning with homomorphic encryption," *IEEE Access*, 2025.
- [19] W. Lin, B. Li, and C. Wang, "Towards private learning on decentralized graphs with local differential privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2936–2946, 2022.

- [20] Z. Ma, J. Wang, K. Gai, P. Duan, Y. Zhang, and S. Luo, "Fully homomorphic encryption-based privacy-preserving scheme for cross edge blockchain network," *J. Syst. Archit.*, vol. 134, p. 102782, 2023.
- [21] K. Sundarakantham, E. Sivasankar, and S. M. Shalinie, "A hybrid deep learning framework for privacy preservation in edge computing," *Comput. Secur.*, vol. 129, p. 103209, 2023.
- [22] G. Xu, J. Zhang, and L. Wang, "An edge computing data privacy-preserving scheme based on blockchain and homomorphic encryption," in *Proc. Int. Conf. Blockchain Technol. Inf. Secur.*, Jul. 2022, pp. 156–159.
- [23] C. H. Nguyen, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, V. D. Nguyen, Y. Xiao, and E. Dutkiewicz, "Encrypted data caching and learning framework for robust federated learning-based mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, pp. 2705–2720, 2024.
- [24] D. Rahbari, M. Daneshtalab, and M. Jenihhin, "Adaptive and efficient federated distillation with selective homomorphic encryption for edge AI," *Expert Syst. Appl.*, p. 131002, 2025.
- [25] B. Kulynych, J. F. Gomez, G. Kaissis, F. du Pin Calmon, and C. Troncoso, "Attack-aware noise calibration for differential privacy," in *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 134868–134901, 2024.
- [26] F. Benabderrahmane, E. Kerkouche, and N. Bouchemal, "Risk-aware privacy-preserving federated learning for remote patient monitoring: A multi-layer adaptive security framework," *Appl. Sci.*, vol. 16, no. 1, p. 29, 2025.
- [27] X. Liu, S. Chen, and Z. Xu, "Privacy-preserving data aggregation mechanisms in mobile crowdsensing driven by edge intelligence," *Electronics*, vol. 15, no. 1, p. 26, 2025.
- [28] <https://github.com/imranbdcse/healthcaredatasets>
- [29] <https://data.mendeley.com/datasets/956gt333hk/2>

Authors' Profiles



Apurva Khandekar, Research Scholar in the Department of Computer Science and Engineering at Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana, India. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Gokaraju Rangraju Institute of Engineering and Technology, Hyderabad, Telangana, India.



Prathipati Ratna Kumar is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana, India.

How to cite this paper:

Apurva Khandekar, Prathipati Ratna Kumar, "A Performance Optimized Privacy-Aware Edge Computing Architecture Using Selective Homomorphic Encryption and Adaptive Differential Privacy", *International Journal of Computer Network and Information Security(IJCNIS)*, Vol.18, No.5, pp. 69-88, 2026. DOI:10.5815/ijcnis.2026.05.05