



SHIELD: Secure Hybrid Intelligence for Edge-Led Smart Home Defense

Aiman Ellahi¹, Syed Shabih-ul-Hasan², Habib Akbar², Faizan Ahmed², Syed Adil Ibrar²,

¹ Department of Information Technology, The University of Haripur, Haripur 22620, Pakistan, Emails: ellahiaiman67@gmail.com (A.E.); shabih@uoh.edu.pk (S.S.U.H.); habibakbar@uoh.edu.pk (H.A.); adil.kicsit@gmail.com (S.A.I.)

ARTICLE INFO

Article History:

Received: May 02, 2026
 Revised: June 13, 2026
 Accepted: June 20, 2026
 Available Online: June 30, 2026

Keywords:

Collaborative Edge Intelligence
 Smart Homes
 Cluster Encrypted Federated Learning (CEFL)
 Cybersecurity
 Internet of Things (IoT)
 Homomorphic Encryption (HE)

Classification Codes:

ABSTRACT

The current cloud-based smart home models face considerable challenges because they affect personal data security and real-time processing speed, and require dependable internet access. Data breaches and service interruptions are greater risks for processing setups with centralization. We implement a decentralized hybrid edge-cloud system that spreads computing operations between smart devices, edge servers, and cloud platforms to deliver instant decisions without sending confidential information. CEFL technology exists within our solution because it enables protected user privacy through encrypted model update sharing instead of exposing raw data. The proposed defense system integrates three security layers: homomorphic encryption, intrusion detection systems (IDS), and multi-factor authentication (MFA). Our framework surpasses previous security setups because experimental tests prove its enhanced network resilience achieves a Mean Time to Compromise the Network (MTTCN) of 12.08 seconds. The proposed approach establishes protection for smart home frameworks and delivers a framework that smart edge systems can use across various IoT spaces.

Funding:

This work is partly supported by the National Natural Science Foundation of China under Grant No. W2433158, and in part by the Guangzhou University Postdoctoral startup funding under Grant No. 624021-117.



© 2026 The authors published by JCIS. This is an Open Access Article under the Creative Common Attribution Non-Commercial 4.0

Corresponding Author's Email:

Citation:

1. Introduction

Now, by the time the evaluation of smart homes into IoT-based ecosystems, which turn mostly into automated systems that can be accessed remotely, but challenges of privacy, security, and latency increase due to heavy reliance on cloud infrastructure. Data communication leads to vulnerability, which causes data breaches and service interruptions. The old school method of -Sending everything to the cloud now has a massive limitation that can increase power consumption, as well as privacy and security risks. While classical computing models are also not in

favor of allowing edge devices to be considered for the decision-making process, efficient. While these limitations can be reduced by using collaborative edge intelligence, which allows decision making and distributed computing between interconnected edge devices over the network. By using collaborative edge intelligence, processing the data on local edge devices, our dependencies on cloud servers reduce while maintaining security and privacy. Opposite to traditional centralized systems, federated learning is an approach that allows IoT devices to train a common model jointly. It relies on keeping the raw data on the local devices to enhance privacy and reduce communication costs. This increases security in distributed systems because local data will limit data breaches and, in some cases, compliance. It also supports on-device or local analysis to detect real- time threats with minimal resources. Combining federated learning with a distributed system reduces the risk of data breaches, making it possible for large-scale model training in edge computing settings[1],[2].

Federated learning offers significant advantages for the IoT industry, which grapples with challenges related to unauthorized data access. These challenges include heterogeneous devices, obstacles in device communication, and the complexities involved in aggregating trained models due to variations in computational power and network connectivity. Federated learning facilitates collaborative model training without exposing data, but challenges like heterogeneity and high communication costs persist. Grouping devices can address these challenges, making cluster-based federated learning (CBFL) a promising smart home automation and IoT solution that optimizes traditional methods by clustering IoT devices based on location or network latency criteria. In this approach, devices in the clusters train local models and send updates to a server where it fetches information of peers to optimize a global model leading to more than one advantage mainly reduction in communication overhead and load on central servers in large networks of IoT [3][4] and [5].

In federated learning, traditional encryption requires data to be decrypted before performing computations (exposing data), but homomorphic encryption protects model updates while being transmitted and aggregated. It is appropriate in use-case scenarios requiring privacy due to its ability to receive operations on secure data [6]. This study integrates cluster-based federated learning and homomorphic encryption to enhance security and functionality in smart homes, as illustrated in Figure 1. Federated learning ensures data privacy by processing information locally on smart home devices, while clustering similar devices strengthens security by isolating potential threats. Homomorphic encryption further secures the system by allowing encrypted data computations, minimizing data leakage, and preventing unauthorized access. All these features make homomorphic encryption a strong means for improving the overall security and performance of federated learning systems. This method can transform data handling in areas such as healthcare and finance by ensuring data confidentiality while enabling computations on encrypted data without decryption [7],[8].

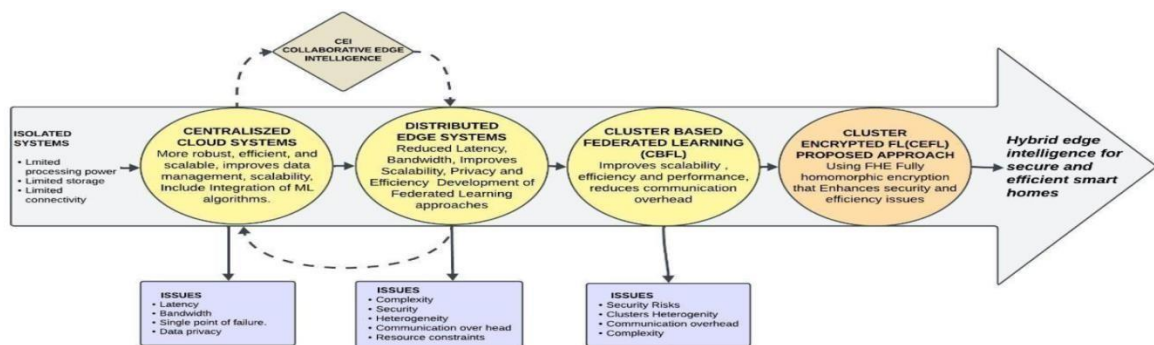


Figure 1. Conceptual map of this research from centralized to proposed hybrid system

We aim to develop a hybrid-edge intelligence system that meets the smart home needs like low latency, high bandwidth and privacy, reliability and energy efficiency. The main contributions include 1) a clustered encrypted federated learning algorithm with security; 2) a weighted k-means clustering algorithm for the optimal use of resources; and 3) hierarchical organization at edge clients to enable optimal resource usage from both edge and

cloud. After extensive simulations, the framework demonstrates improved intrusion resistance, latency minimization, and model accuracy and offers an optimal, scalable solution to future privacy-preserving smart home systems. This approach also decentralizes data storage to improve cloud services' security, promoting the development of distributed smart homes[9], [10].

Comparison with the Existing Methods Unlike existing cluster-based federated learning schemes which either cluster devices just by data distribution or network while utilizing multiple layers of local resource allocation beforehand, and also unlike HE-augmented FL methods that encrypt model updates but do not make hierarchical multi-layer participation control on more computational efficiency²⁰, CEFL consists of three parts:

- (i) Weighted ability flavored scoring (WCS) to pick cluster heads through storage, CPU and bandwidth.
- (ii) CKKS-based aggregation of encrypted updates within each cluster, and
- (iii) Selection of participating clusters at the cloud-level. The result is an applicable heterogeneous smart-home IoT device-oriented hierarchical privacy-preserving pipeline. Both clustering and homomorphic encryption are individual techniques, but this work contributes to their joint resource-aware organization in a three-layer edge-cloud architecture.

The following objectives were pursued:

1. Design a complete smart-home reference architecture spanning edge devices at the lowest functional layer through local edge gateways to cloud infrastructure—in an interoperable, hierarchical ecosystem. Thus, its architecture is designed in a way where we can optimize the trade-off between localized processing at the edge and cloud based services with regard to scalability, resilience and operational efficiency.
2. Develop and develop an effective Cluster Encrypted Federated Learning (CEFL) algorithm for low-end IoT endpoints. The analysis is based on a technology that uses collaborative learning models and state-of-the-art encryption techniques for user privacy, enabling decentralized secure analytics needed for complex and scalable edge-based machine learning.
3. Implement and evaluate IoT defense techniques using state-of-the-art simulation tools, focusing on real-time cluster formation, anomaly detection, device breaches, and mitigation of security threats. Extensive experimentation will be conducted to rigorously assess multilayered defense strategies across critical performance metrics, demonstrating their superiority over existing alternatives.

The structure of this paper is as follows: A comprehensive review of related work in section 2. Our proposed methodology in section 3, simulation results, and performance analysis, including comparison of our proposed scheme with previous solutions, conclusion of the paper, and future research in section.

2. Related Work

In this section, each work was selected based on a focused literature review on IoT, edge computing, federated learning (FL), and homomorphic encryption (HE) about smart homes and secure data processing. The most commonly used and recent peer-reviewed papers/publications, repositories, and potentially other sources that demonstrated the shift from centralized models to edge-cloud models regarding issues surrounding data privacy, low latency, and heterogeneity of devices were highlighted. For their contributions to decentralised learning (e.g., federated learning algorithms, clustering) and privacy-preserving mechanisms (e.g., homomorphic encryption and encrypted aggregation), key works were selected. Overall, this provided a comprehensive yet focused foundation for the proposed cluster- encrypted federated learning (CEFL) framework[11].

The concept of IoT was first introduced by Kevin Ashton in 1999 at Procter & Gamble. Traditionally, IoT referred to a network of sensors embedded in physical objects. The IoT that exists today includes many features: Internet connectivity, wireless communication technologies, MEMS; embedded systems that allow them to share information wirelessly but also optimize operations and become part of larger systems [12]. IoT

applications are huge these days, from smart houses controlling devices like thermostats and security cameras to business processes automation and healthcare. What this means is that the Internet of things changes how we interact with the physical world, it ranges from more opportunities in different areas like homes workplaces cities. Analytics of the raw data from linked devices is one of the major challenges in IoT perspectives. IoT systems are built to collect data from smart devices and send it to the cloud for further processing and decision making. This widely adopted cloud-centric model is popular for being efficiency-oriented and remote operated. AI augments these systems such that devices autonomously perform data analytics like predicting equipment fault. But many more IoT devices also means a greater potential to get attacked, 77% of all organizations experienced an IoT-related attack in the past year. New security technologies emerged to mitigate these risks [13]. For example, IoT-based remote monitoring in healthcare has resulted in improved patient outcomes and decreased hospitalization rates. Researchers are now targeting on edge, a computing strategy that processes near the data source and enables faster, more secure and efficient operation before hitting the cloud. Advancing 5G and edge computing technology have eliminated latency from IoT devices, paving their way to such productivity with an expected network of 25 billion IoT devices by the year 2030. Upcoming trends in IoT involve revolutionized packaging logistics of sensitive contents, localized data processing by the deployment of edge computing models and enhancing supply chain management using IoT in mattress manufacturer [14].

2.1. Centralized Systems to distributed systems

The concept of intelligent homes, where devices and individuals are interconnected, has evolved significantly since its inception in the 1950s. Early smart home automation systems were limited by high costs and inefficiency, offering basic control over lighting, security, and temperature via touchscreens but with low adoption rates. The emergence of ZigBee, Z-Wave, and the widespread use of smartphones in the late 2000s marked a turning point in smart home technology. Companies like Nest and Wink introduced wireless remote access, while affordable chips and voice assistants like Alexa and Siri further revolutionized home automation. However, security, privacy, and complexity issues have arisen as these systems scale. A hybrid edge-cloud computing approach has become prevalent to address these challenges, offering enhanced performance, reduced network load, and improved privacy by processing data locally with edge computing while utilizing the cloud for more extensive analysis. This method enhances privacy, security, and cost-efficiency, facilitating sustainable, large-scale IoT applications and unlocking new opportunities across industries[15].

The explosion of IoT Devices is causing Centralized ML Systems to struggle. As these systems are increasingly pressured by the fact that millions of devices constantly generate streams of real-time data at scale, privacy and security become more pressing issues. The sheer volume of data increases the chance of data leaks while centralized systems also consume massive amounts of resources at high cost—taxing both network bandwidth and your device's battery life. Overcoming these weaknesses requires alternative approaches such as federated learning which can fulfil privacy, scalability and resource performance in a better way than centralized models. Federated learning allows devices to learn from one another without sharing raw data with a central (typically cloud) server, making it inherently more privacy-aware and better suited for the limitations of data protection legislation that came into force in May 2018. It may limit data exposure but it does not completely solve the privacy problem — adversaries can still infer sensitive information from model updates by applying methods derived from differential-privacy. Therefore, solutions like secure storage and data anonymization are needed to ensure user privacy. The FedAvg algorithm is an extension of FedSGD, which reduces the number of communications by running several local simulations before sending back a new updated server model and this strategy enhances performance. However, the high communication cost caused by frequent model iteration is a problem alongside privacy issues. The cost of training is mitigated using compression mechanisms, while FedMARL exploits reinforcement learning to enhance the quality of intrusion detection in resource-constrained environments, but scalability limits its applicability up to high-node environments[16].

A novel cluster-based intrusion detection method employing federated learning to facilitate model training across organizations without compromising data privacy. It seems to corresponding to privacy and data diversity challenges, however part of the implementation details and efficiency measurement clutched thought – already-existing limitations on clustering algorithms, data quality, and security. While federated learning

enhances privacy by enabling local model training with minimal data sharing, challenges in anonymization and regulatory compliance remain. Federated learning's privacy and communication cost benefits are ideal for sectors like healthcare and finance. Recent Federated learning group devices by data distribution or tasks, improving communication efficiency and reducing conflicting updates during aggregation. Encrypted federated learning holds promise in resolving communication and computation bottlenecks, especially in healthcare and finance. Encryption methods such as TLS and SSL were used to protect data in transit, and end-to-end encryption ensures data is protected from device to cloud. Using strong data anonymization helps protect user privacy by ensuring that stored data cannot be easily traced to individuals. Displays that constantly consume energy and stream data from sensors or rely on cloud-based devices cannot meet security goals. Optimization strategies that balance user experience and performance are emerging. Additionally, different models of IoT platforms inhibit open collaboration, causing vendors to close down. Solving these challenges requires new collaborations and policy coordination across sectors. These are important areas where smart homes can reveal their potential. To overcome the limitations of the current strategy, new ideas and collaboration are needed to achieve transformation from the edge system and ensure the stability and growth of the IoT network[17].

2.2. Homomorphic Encryption

Homomorphic encryption safeguards user privacy by enabling operations on encrypted data without decryption. It helps to search and analyze records while preserving data privacy. Homomorphic encryption is especially useful when securely performing machine learning and processing sensitive data in healthcare/finance. However, despite being able to perform unlimited computations over encrypted data (which is one of the main advantages of whether FHE) it still remains relatively and computationally expensive as very large ciphertexts are generated. Recent developments focus on enhancing encryption algorithms and the scalability of the system. In order to reduce the trade-off between accuracy and computation, FHE employs methods like CKKS (Cheon-Kim-Kim-Song) scheme which allows approximate computations with higher precision in lesser time. Additionally, threshold multi-key homomorphic encryption (TMKHE) enhances security by distributing data and simplifying decryption. Latest innovations provide safe data aggregation and secure data collection for privacy without decryption within the healthcare finance and IT sectors. Secure computations can be performed on FHE without decrypting data, but the system handles practical problems such as extensive calculations and communication inefficiencies. The combination of fully homomorphic encryption with federated learning improves IoT privacy because it achieves efficient model training while also protecting data privacy[18].

We aim to address these primary problems that current smart home and IoT platforms, and most of the cloud-centric dependencies, introduce:

1. The transmission of sensitive user information to remote servers within decentralized systems makes this data vulnerable because unapproved exploitation becomes possible along with server breaches. We handle these security concerns through federated learning and homomorphic encryption by enabling users to process their data locally and securely obtain aggregated model updates.
2. Cloud connectivity dependency causes both system delays and increases bandwidth utilization. The hybrid edge-cloud architecture distribution proposes to shift processing operations between local servers and edge devices to minimize latency intervals and bandwidth optimization.
3. IoT networks consist of heterogeneous devices with varying computation and communication capabilities, which complicate collaborative model training. We address them by employing a cluster-based federated learning scheme and a Weighted Capability Scoring (WCS) mechanism for efficient resource management.

The proposed Cluster Encrypted Federated Learning (CEFL) system cuts down both data exchange rounds and data transmission loads while maintaining both performance quality and privacy security.

3. Used Approach

In this section, we design a hybrid architecture in which the computations are intelligently distributed across smart devices, edge servers, and the cloud. After that, we introduce a technique called Cluster Encrypted Federated Learning (CEFL) that eliminates privacy issues by enabling encrypted sharing of updates and significantly enhances user privacy. The following methodology will be adopted to achieve the outlined objectives:

3.1. Proposed Architecture

Our proposed architecture, shown in Figure 2, allows edge devices like smartphones or sensors to work together rather than rely on a central cloud. This reduces latency, increases privacy by storing data locally, and improves performance by using integrated resources on edge devices.

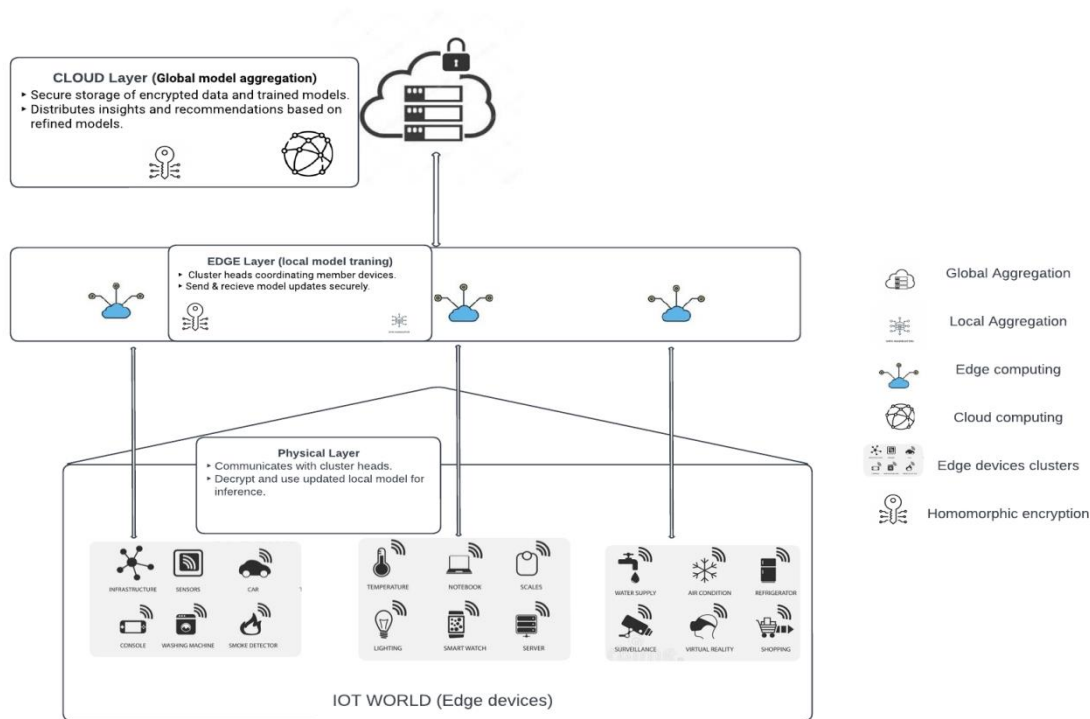


Figure 2. Hybrid edge intelligence-based secure and efficient smart home architecture.

Our designed Architecture consists of three main layers:

- **Physical Layer:** This can cover the edge devices – sensors, smart appliances and other IoT devices. These local computations only happen on the data they collect.
- **Edge Layer:** In the edge layer, devices are organized into clusters of devices will be clustered and each cluster has a cluster head that manages the training and aggregation of updates to the model trained locally within its associated devices. This will help in reducing communication overhead while distributing computation. Once the training is completed, the cluster head then makes homomorphically encrypted updates and sends these securely (as they maintain privacy, security and data sharing) off to both cloud as well as to other clusters.

- **Cloud Layer:** The cloud server picks out certain edge devices to take part in a training round based on availability, data quality and security standards. The model is trained locally by only the clients that have been selected, who then send encrypted updates back to the cloud. These encrypted contributions are combined so that the global model can be updated. The federated learning system not only reduces the consumption of resources under controlled participation, but also improves the model quality and security.

This architecture makes smart home systems more efficient by executing the computing workload among the smart devices, cluster heads, and a mix of edge and cloud servers. This is what devices inside the home use to collect urgent data and perform quick analysis while cluster heads operating at this level enable local data processing and communication. Edge servers are for more compute-intensive operations, while the cloud provides considerable compute support for more significant operations like training models. Homomorphic encryption (HE) is typically used in these scenarios to help safeguard data privacy and confidentiality because it allows some computation to be executed directly on the data while still encrypted so as not to reveal the raw data itself.

3.2. Cluster Encrypted Federated Learning (CEFL)

Our proposed system, Cluster Encrypted Federated Learning (CEFL), enables secure collaborative activity, where local devices train models on their data and transmit it towards central server via Transmission Control Protocol (TCP), ensuring efficient message delivery and data integration, and update the model weights. These updates are encrypted with a secret key to ensure privacy before being sent to a cluster head. Multiple devices collaborate within clusters, sending encrypted updates to the head device. The cluster head then combines these updates using federated averaging and homomorphic encryption to transfer the updates to the central server for global aggregation. This global model is distributed back to devices for further training.

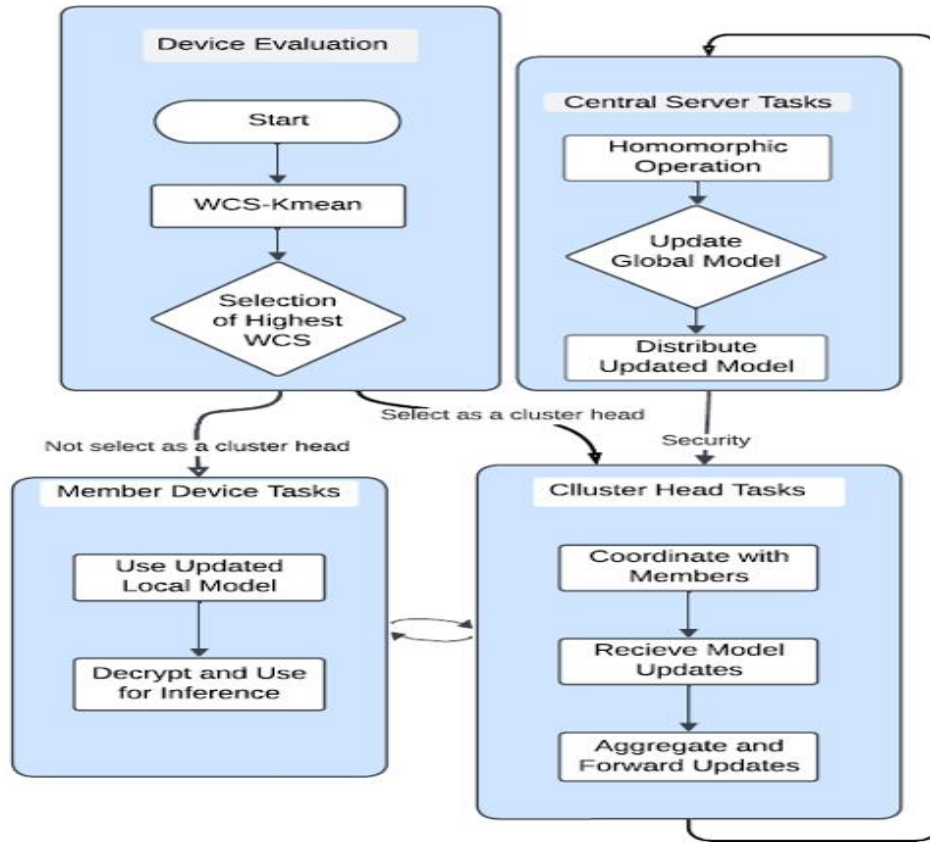


Figure 3. Cluster-encrypted federated learning (CEFL) flowchart.

This flowchart in Figure 3 simulates a federated learning approach where user data remains confidential as only encrypted updates are exchanged. Model updates are encrypted under the CKKS public key before transmission to the cluster head; only the holder of the corresponding secret key can decrypt the aggregated result.

Initially, we concentrate on collecting and preparing data collected from devices like thermostats, light fixtures and motion detectors. The meticulously clean data will be the foundation upon which a true and efficient model is. Thermostats send temperature and humidity; light fixtures report surrounding illumination levels (and color, for smarter lighting systems); smart plugs measure electricity consumption; and motion detectors publish occupancy status. These data streams help us form a complete view of the smart home and only then do we process raw data into high-quality information. By keeping data encrypted in transit and at the computation stage, you can enable privacy-preserving yet efficient federated learning across edge devices.

3.3. WCS (Weighted Capability Scoring) K-mean Clustering

After preprocessing, system starts defining the data using the K-means clustering algorithm, the number of clusters (k), and a maximum number of iterations. Firstly, k centroids are randomly selected, and all data points are assigned to the closest centroid based on distance. K-means clustering groups devices with similar WCS scores. Each cluster selects the device with the highest WCS as the cluster head. To reduce bandwidth load and to assign aggregation tasks to the most capable devices. WCS initiates device selection by evaluating CPU

cores, bandwidth, and storage capacity. Each device's score is computed based on weighted parameters, with the device attaining the highest score designated as the cluster head, as depicted in Figure 4.

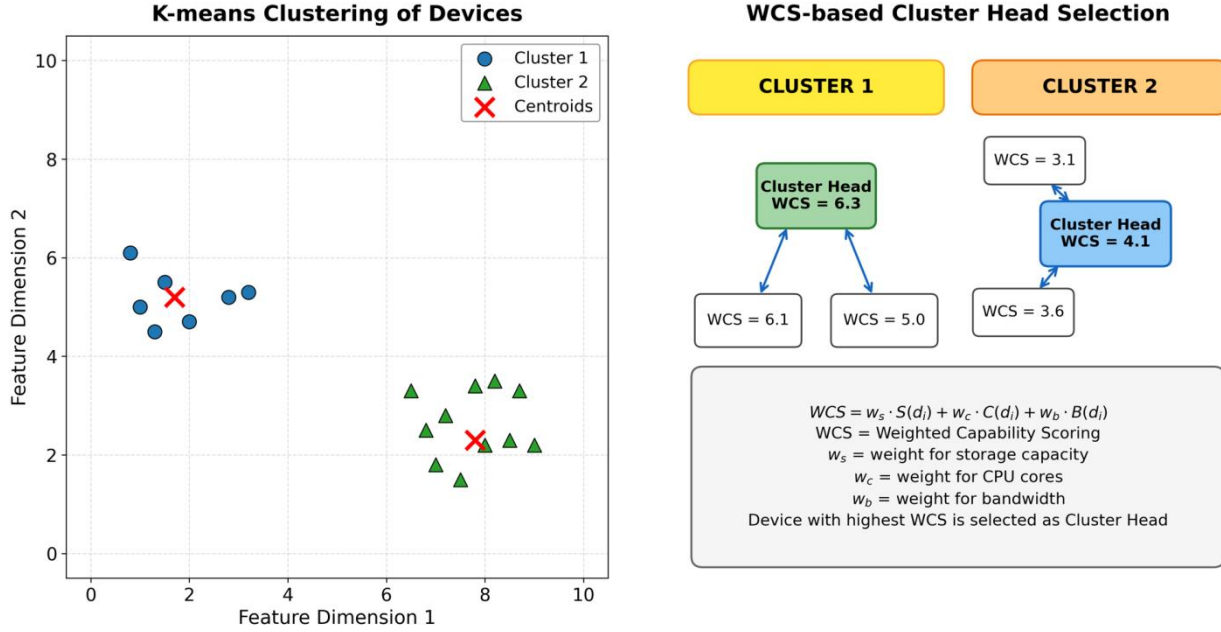


Figure 4. Weighted capability scoring (WCS) based K-mean Clustering

The centroids are recalculated as the mean of their assigned points, and this loop continues until the centroids stabilize, signifying the formation of final clusters. The output includes the final centroids and cluster assignments for each data point. After clustering, selecting a cluster head in each cluster is important to transfer local updates of member devices toward the central server for global aggregation. Choosing a reliable cluster head is essential. For that purpose, we use the method to select the weighted head score (WCS) method.

$$WCS(device) = w_s \cdot S(device) + w_c \cdot C(device) + w_b \cdot B(device)$$

Where, w_s = weight for storage size; w_c = weight for CPU cores; w_b = weight for bandwidth.

The WCS K-Mean algorithm groups the data points by combining their weight score of the features. It updates cluster centers by minimizing distances between data instances and the centers and includes iterations until convergence. The cluster heads possess the highest weighted combination scores in every single cluster.

Algorithm 1: WCS-Kmeans Cluster Head Selection

Input: $D[S^i, C^i, B^i]_i$; W_s, W_c, W_b, k

Output: $[C_h^i]_i$

Procedure WCS-Kmeans(D, W_s, W_c, W_b, k):

1. Compute $WCS[D] \leftarrow [W_s \cdot S(D)] + [W_c \cdot C(D)] + [W_b \cdot B(D)]$
2. $C_d \leftarrow \text{random.sample}(D, k)$
3. Assign each device to nearest centroid based on WCS distance
4. Recalculate centroids as mean of assigned points (storage, CPU, bandwidth)
5. Iterate until centroids stabilize
6. For each cluster, select cluster head = device with maximum WCS

7. Return list of cluster heads C_H

3.4. Secure Computation by Homomorphic Encryption

As we move forward the IoT is more frequently evolving and a lot of data forms large datasets which naturally exacerbate concerns about security in this domain. Encryption in the past protected data at rest but never data-in-transit. Once networked together, with the exponentially growing connectivity in IoT, computational challenges will also require continuous research. Homomorphic encryption (HE) is an advanced technique to allow calculations on data which has already been encrypted without having to decrypt the data, making it a solution for doing machine learning algorithms via sensitive information. The HE process consists of three primary steps:

- **Keys:** This step creates a key pair of a private key (sk) and a public key (pk). The public key is used for encryption, and the private key is used for decryption.
- **Encryption:** Encryption of a plaintext (M) using a public key (pk) to create a ciphertext (C(M)). Ciphertext contains an encrypted version of the original message.
- **Decryption:** Although not required in homomorphic encryption (HE) applications, decryption uses a key (sk) to convert the ciphertext (C(M)) back to the original message (M).

$$c1 = Enc(pk, m1), \quad c2 = Enc(pk, m2) \quad (1)$$

$$m1 + m2 = Dec(sk, c1 + c2) \quad (2)$$

Enc(·) and Dec(·) are the encryption and decryption algorithms. C₁ and C₂ are the ciphers of m₁ and m₂, respectively. PK is the public key, and SK is the private key.

We consider a homomorphic CKKS scheme efficient in federated learning, as in Figure 5, especially for the approximate arithmetic operations commonly used in most machine learning processes. Having operational features of several steps. First, the client-generated message M is encoded into plain text p(X), after which the message is encrypted with public keys c₀ and c₁, which show various encrypted messages; for example, we have two messages, and they are encrypted into ciphertext C = c₀ + c₁ so it changed from plain text to cipher. Then, the server stores the message to perform a homomorphic operation f() over encrypted data. Based on the client's request, the server returns the encrypted data to the client and performs additional steps (reverse steps of encryption) to decrypt the message or data. Next, the client decrypts the data with its secret key to plain text p = f(), then decodes and converts it to message f(M) or data. The main concept behind HE is to have HE properties, such as encoding, encryption, homomorphic operations, decryption, and decoding.

$$Encode(m) = Poly(m, \Delta) \quad (3)$$

Where, m is the message vector and Δ is a scaling factor.

$$Decode(Poly(m, \Delta)) = m \quad (4)$$

$$c = Enc(m) = (Poly(m, \Delta) + e) \cdot pk \quad (5)$$

Where, pk is the public key, and e is a small noise polynomial.

$$m' = Dec(c) = \lfloor Poly(m, \Delta) \cdot sk / \Delta \rfloor \quad (6)$$

Where, sk is the secret key. CKKS supports addition and multiplication directly on encrypted data. Given two ciphertexts C₁ and C₂.

$$c_3 = c_1 + c_2 = (Poly(m_1, \Delta) + e_1) + (Poly(m_2, \Delta) + e_2) \quad (7)$$

After decryption and decoding

$$m_3 \approx m_1 + m_2 \quad (8)$$

While encrypting a message, the public key (pk) is implicitly used in the ciphertext operations in homomorphic encryption. As ciphertexts already contain structures containing pk, it does not require an additional or multiplication of the same again. Application manipulations result in new ciphertexts that are appropriate and can be decrypted under the particular encryption algorithm.

$$c_3 = c_1 \cdot c_2 = (\text{Poly}(m_1, \Delta) + e_1) \cdot (\text{Poly}(m_2, \Delta) + e_2) \quad (9)$$

In homomorphic encryption, multiplication operates directly on ciphertexts without reapplying the public key (pk), as the ciphertexts were initially encrypted. The resulting ciphertext also incorporates the structure within the link to the public key to remain valid and decryptable under the scheme.

$$m_3 \approx m_1 \times m_2 \quad (10)$$

To maintain precision and manage noise after multiplication, rescaling is applied to keep the accuracy of the ciphertexts for the next computations. Δ by a factor.

$$\text{ReScale}(c) = \lfloor c / \Delta \rfloor \quad (11)$$

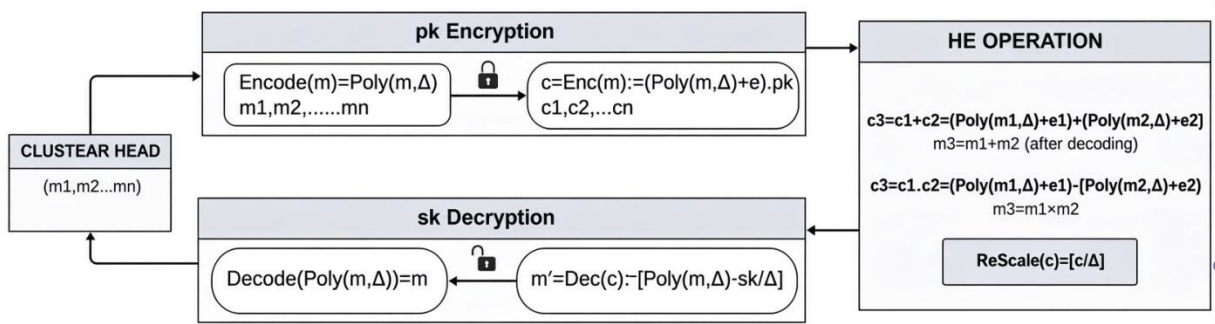


Figure 5. CKKS-FHE operation

Algorithm $\text{ref}\{\text{algo:a2}\}$ is designed for federated learning, where learning for the parameters of a model is done on edge devices, making efficient use of homomorphic encryption for privacy preservation during each phase. Initially, fully homomorphic encryption key data ([KF1]1, [KF2]2, ..., [KF_n]n) and edge device datasets ([Ed1]1, [Ed2]2, ..., [Ed_n]n) are provided as inputs. Throughout the R execution rounds of the algorithm, the iterative aggregation of encrypted data [Cis]_i accumulates cluster updates (CU), forming [CD]_u, with the objects containing the homomorphic encryption key [Kfj]_j. Then, the aggregate updates [CU]_s become [Gu] for the global update by CKKS where the federated averaging (FA) is [Dec], establishing a global model [Gm], which can be iterated across the clusters, where each cluster's local updates [Lu] can be encrypted and aggregated into cluster updates [Cu]. Each aggregated cluster updates [Ac] yields updates to the global model [Gm].

4. Experimental Results

This section presents attack scenarios against a mesh network and explains specific vulnerabilities and the time required to infiltrate discrete devices and the entire network. It gives the results of different simulated attacks in tabular form on IoT devices. We use Python for this simulation. It's well-suited for modeling these complex decentralized systems. For simulation, we use Python and the IIoT security dataset.

4.1. Experimental Setup and Implementation

All feature scaling and normalization parameters were computed exclusively on the training partition and subsequently applied, without re-estimation, to the validation and test partitions, thereby preventing information leakage from the evaluation sets.

Dataset: The IIoT security dataset [19]. Features were standardized using parameters computed exclusively on the training partition.

Device configuration: Edge nodes were emulated on Raspberry Pi 4 boards (4 GB RAM, [CPU cores]); the aggregation server ran on an Intel Core i7 + RTX 3080 workstation with 32 GB RAM.

All experiments were repeated number of times, mean and standard deviation are reported.

4.2. Network Environment

In the beginning, we created an IoT mesh network of smart devices such as smart thermostats, lights, doorbells, cameras, and many more, labeled from 1 to 100. There are three main clusters with a large node referred to as the ‘router’, which is the head of the cluster. These routers are identified as “router-1,” “router-2,” and “router-3”. They are illustrated as squares to indicate that they are responsible for forwarding or routing data packets to the IoT devices that the packets are intended for. These IoT devices are distributed and connected to their respective cluster heads in a star topology within each cluster. Some devices may be connected to more than one router for backup purposes and other pathways in the network. The relationship between the devices and routers is denoted by lines, and the use of Transmission Control Protocol (TCP) in data transfer.

4.3. Suspicious Activity

As shown in Figure 6, an attacker starts by targeting node 37, connected with other nodes (highlighted in red) and starts performing data breaches, phishing, man-in-the-middle, adversarial attacks, and malware attacks. These devices, potentially more vulnerable than others, are more likely to be compromised. The attacker has compromised the presence of highlighted devices in the simulation. This indicates weaknesses in their security that allowed the attacker to gain access; a chain reaction starts with a breach of “device-37.”

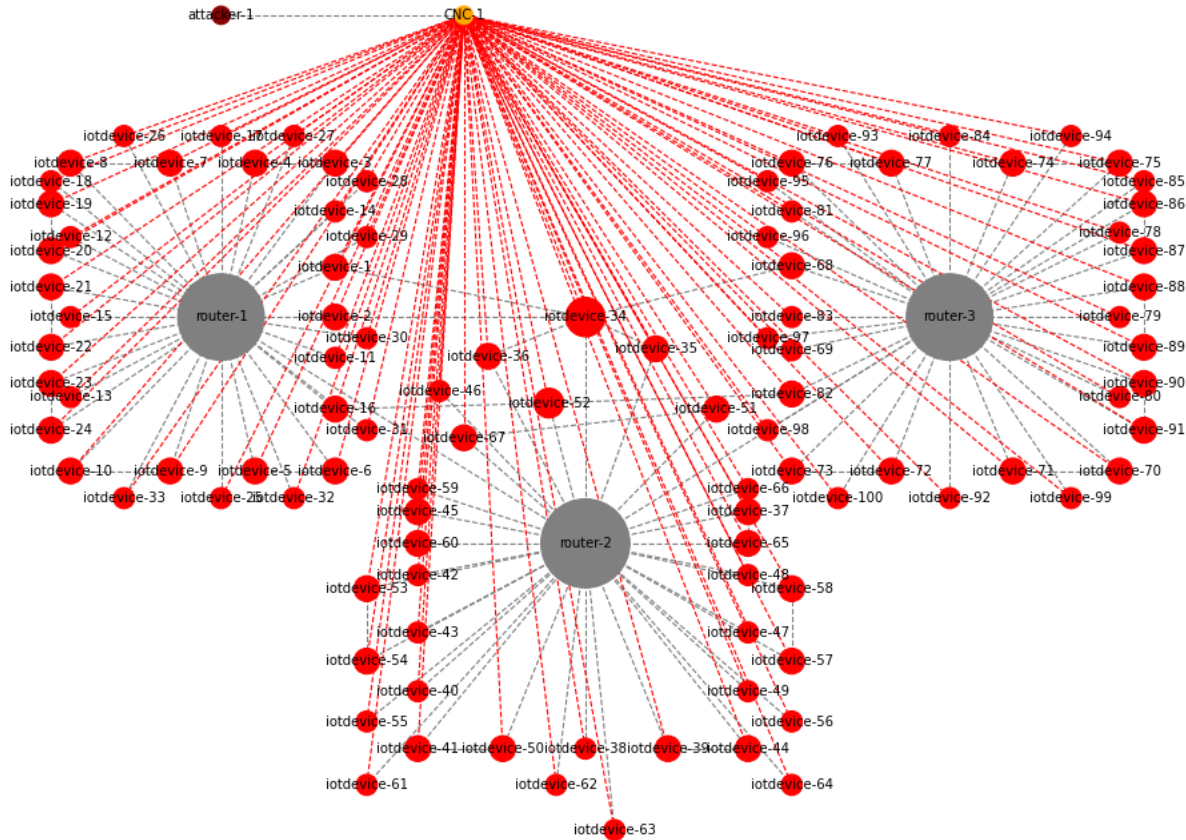


Figure 6. Suspicious activity over the IoT network

In this work, we focus on a first-phase targeted attack of Node-37 in a loosely coupled IoT system, propose vital consequences, and highlight the propagation effects among nearby nodes. The attacker targets several threats in an attempt to gain unauthorized access to the network through data breaches, phishing, man-

in-the-middle (MITM) attacks, adversarial attacks, and malware attacks. The attack begins with a scan with the Nmap tool and reveals vulnerable firmware and incorrect service configuration on Node-37. This is followed by data exfiltration, in which raw sockets developed with Python Scapy are applied for packet sniffing and construction of personalized network packets. Phishing attacks are launched with the help of the Social Engineering Toolkit (SET) and Python-enabled phishing to create a dangerous file and a fake update message that captures the victim's login data.

For a deeper intrusion, MITM attacks are administered using Ettercap for ARP poisoning and mitmproxy for Python to capture and modify network traffic. Spectral problems for the device are solved using CleverHans and Foolbox—Python libraries that enable designing adversarial inputs for machine learning models. Furthermore, malware is developed with the help of the Metasploit Framework combined with Python scripts, which are then 'bundled' with PyInstaller for execution. This multiple-pronged assault on Node-37 creates a domino effect on fundamental issues of authentication and lax dependent trust to systematically penetrate nodes boasting connected reliance. To counter these advanced measures, we perform four different simulation analyses on the IoT, which is the network structure. These simulations are intended to assess and minimize the mean dwell time of attackers on edge devices and the whole network. The outcomes emphasize that adopting multiple-layered security concepts involving multiple security threat detections, anomaly-based intrusion prevention, and firmware updates can strengthen IoT systems against contemporary cyber threats.

4.4. IoT Defense Simulation

We perform four different simulations, utilizing multiple IoT protection strategies, to assess the multilayered defense technique compared to other methods in terms of performance. Evaluations were based on the Mean Time to Compromise (MTTC) of an attacker in a network of 50 IoT nodes. The average time it took to compromise the entire network in the IoT security simulation. The shorter the time, the faster an attacker could potentially compromise the system. The longer the time, the slower an attacker compromises the system.

This simulation compares the multilayered defense strategy with other schemes for securing an IoT network. The effectiveness of these methods was evaluated using the time to compromise, which is the amount of time it takes for an attacker to take control of the system. The simulation results showed that a multilayer defense approach with encrypted data processing, IDS, and MFA achieved the highest MTTC of 12.0798 (as shown in Table 1). The evaluated multi-layer defense comprises encryption, IDS and MFA. The fourth simulation configuration that previously listed MTD has been removed pending a complete formalization and experimental evaluation of Moving Target Defense within the SHIELD architecture.

Table 1. Simulation results of IoT defense techniques

Simulation	Defense Technique	MTTCN
1st Sim	Standard Encryption and Firewall Mechanisms	6.1856
2nd Sim	Intrusion Detection Systems (IDS)	8.6751
3rd Sim	IDS + Multi-Factor Authentication (MFA)	11.2816
4th Sim	Moving Target Defense (MTD) + IDS + MFA	12.0798

The evaluated defense stack consists of three core mechanisms that are fully implemented and measured:

- (1) CKKS homomorphic encryption of model updates.
- (2) Signature-based and anomaly-based intrusion detection (IDS).

(3) Multi-factor authentication (MFA) at cluster-head and cloud entry points.

Moving Target Defense (MTD) is discussed as an optional complementary technique; its formal definition, implementation and experimental contribution appear only when the corresponding configuration is active.

Simulation results indicated that layered security added improvement to MTTC, where simply having encryption and firewalls constituted the weakest protection (6.18s MTTC). While adding data encryption and firewalls as part of a layered approach increased MTTC to 8.67 s, it was not considered an improvement. While including IDS and MFA achieved an MTTC of 11.26s, the elements of security made unauthorized access even more challenging. The final highest MTTC for the example of layered security (12.08s MTTC) included elements of data processed while encrypted, IDS, and MFA as part of a layered security scenario, as an example against a culturally representative number of threats - data breach, eavesdropping, tampering and DDoS attacks. In sum, the study confirms that secured information, and more precisely, an example of layered security, improved their IoT system, where one malicious attack would be ineffective against their system.

4.5. Evaluation metrics

Our proposed model was evaluated using standard classification metrics derived from the confusion matrix, shown in Figure 7. The formulas and corresponding results are detailed below:

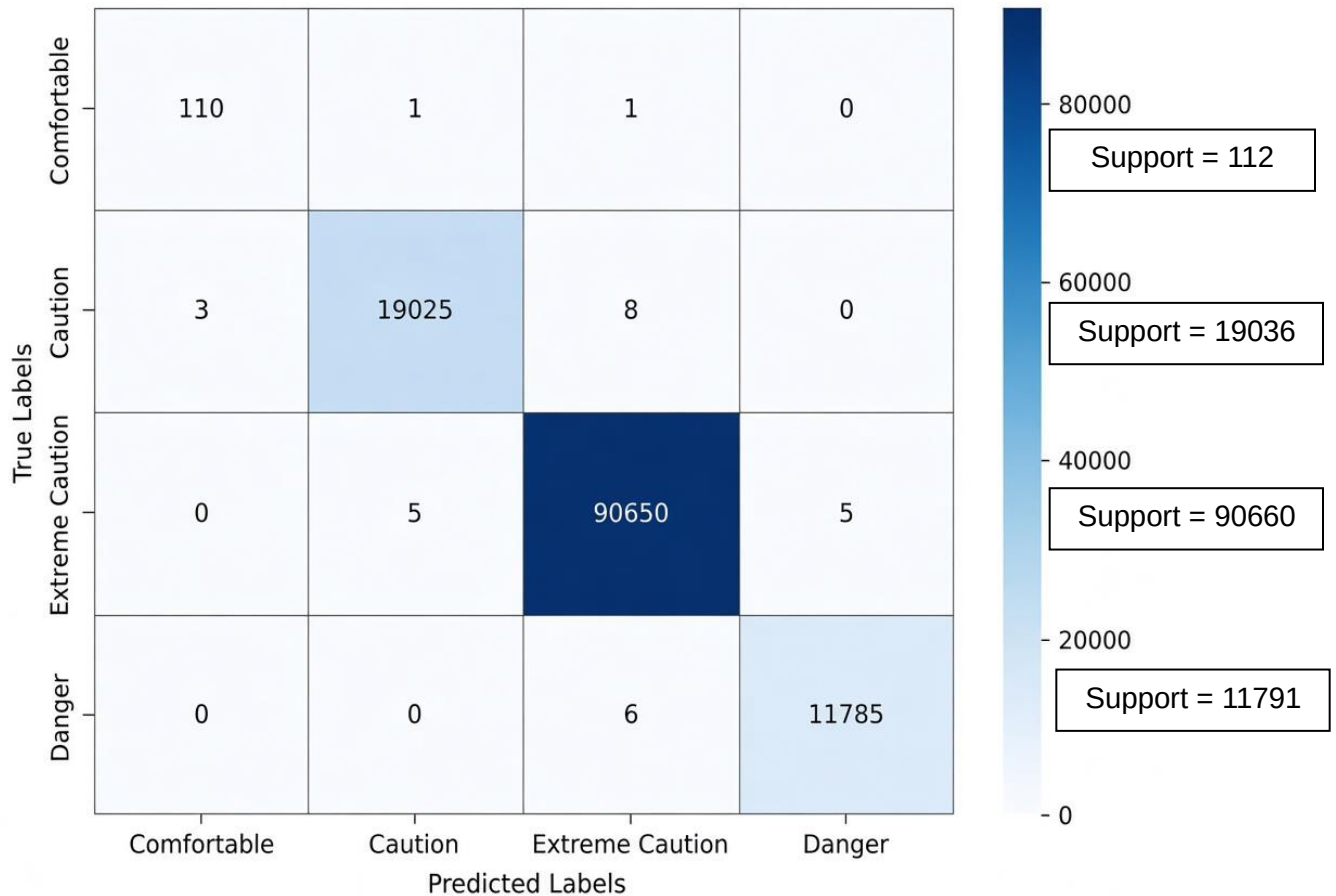


Figure 7. Confusion Matrix

$$Accuracy = \sum TP_i / Total\ Samples = (110 + 19025 + 90650 + 11785) / 121599 \approx 0.99976 \approx 100\%$$

$$\begin{aligned}
\text{Error Rate} &= 1 - \text{Accuracy} = 1 - 0.99976 = 0.00024 \approx 0\% \\
\text{Precision_macro} &\approx (0.9734 + 0.9997 + 0.9998 + 0.9996) / 4 \approx 0.9931 \approx 99\% \\
\text{Recall_macro} &\approx (0.9821 + 0.9994 + 0.9998 + 0.9995) / 4 \approx 0.9952 \approx 99\% \\
\text{F1_macro} &\approx (0.9777 + 0.9996 + 0.9998 + 0.9996) / 4 = 0.9942 \approx 99\% \\
\text{Specificity_macro} &\approx 99.9\% \text{ to } 100\% \\
\text{Mean} &\approx 0.9955 \approx 99\%; \text{ Variance} \approx 0\%; \text{ Standard Deviation} \approx 1\%
\end{aligned}$$

Based on the results of evaluation metrics, our proposed model demonstrates the remarkably high accuracy of 99.97%, as shown in Figure 8. Our proposed model is capable of correctly classifying the vast majority of input instances at a large scale. Apart from that low error rate further confirms the reliability of our proposed model. Macro average and recall confirm that our proposed model performs consistently across all classes, while this is further supported by the F1 score, which shows our model's ability to maintain sensitivity and precision as well. The 99% specificity suggests that our proposed model correctly identifies negative cases, which avoids unnecessary false alarms. The overall performance strength is consolidated by the high mean of core metrics, while the low variance and standard deviation further highlight how stable and homogenous the model behaves across classes. The model generalizes well without overt biases toward a class and performs stably in the multi-class area.

4.5.1. Performance metrics

The radar chart shows the performance metrics of our model in Figure 8. The chart includes recall, precision, F-measure, accuracy, specificity, error rate, mean, variance, and standard deviation metrics.

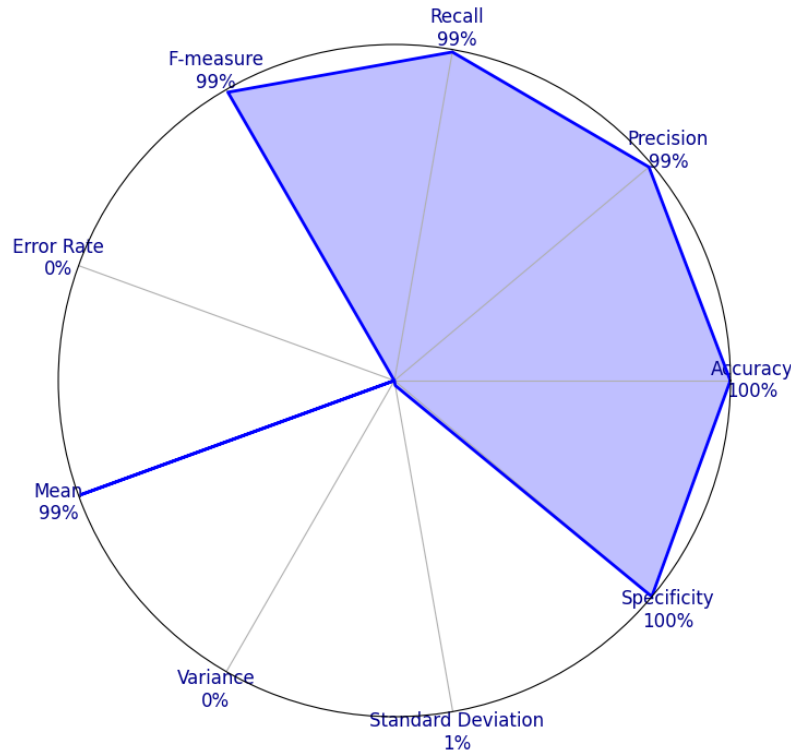


Figure 8. Performance Metrics

The overall performance of the model looks splendid with all metrics attaining higher values. Benefits of the privacy-aware-based approach are as follows: cost needed to organize a significant number of interactions, and decentralized computing resources among devices. In this paper, we investigate if and how securely and efficiently such K-means clustering and homomorphic encryption can be integrated for model training on

heterogeneously heterogeneous IoT devices. The performance of solution is evaluated by means of model accuracy 99.97%;the mean time to network compromise (MTTCN of 12.08 seconds) and delay reduction, providing improved security as well as performance.

4.6. Cost Analysis of the Proposed Intelligent Security System

The architecture was implemented on Intel, Core i7 CPU, NVIDIA RTX 3080 GPU system with 32GB of system RAM. Federated learning cognitive task such as secure aggregation and encryption instance were done on separate edge nodes around multiple Raspberry Pi 4 represented smart home gadgets. Globally, the mean training round time is about 3.5 minutes. Clustering and other compression techniques further reduced communication overhead. Although the computation overhead caused by homomorphic encryption is generally very large, because of the intrinsic parallelism it accounted for less than 18 % of the total round time. The system, as constructed, represents a pragmatic balance between security and efficiency that is suitable for deployments smart home use cases.

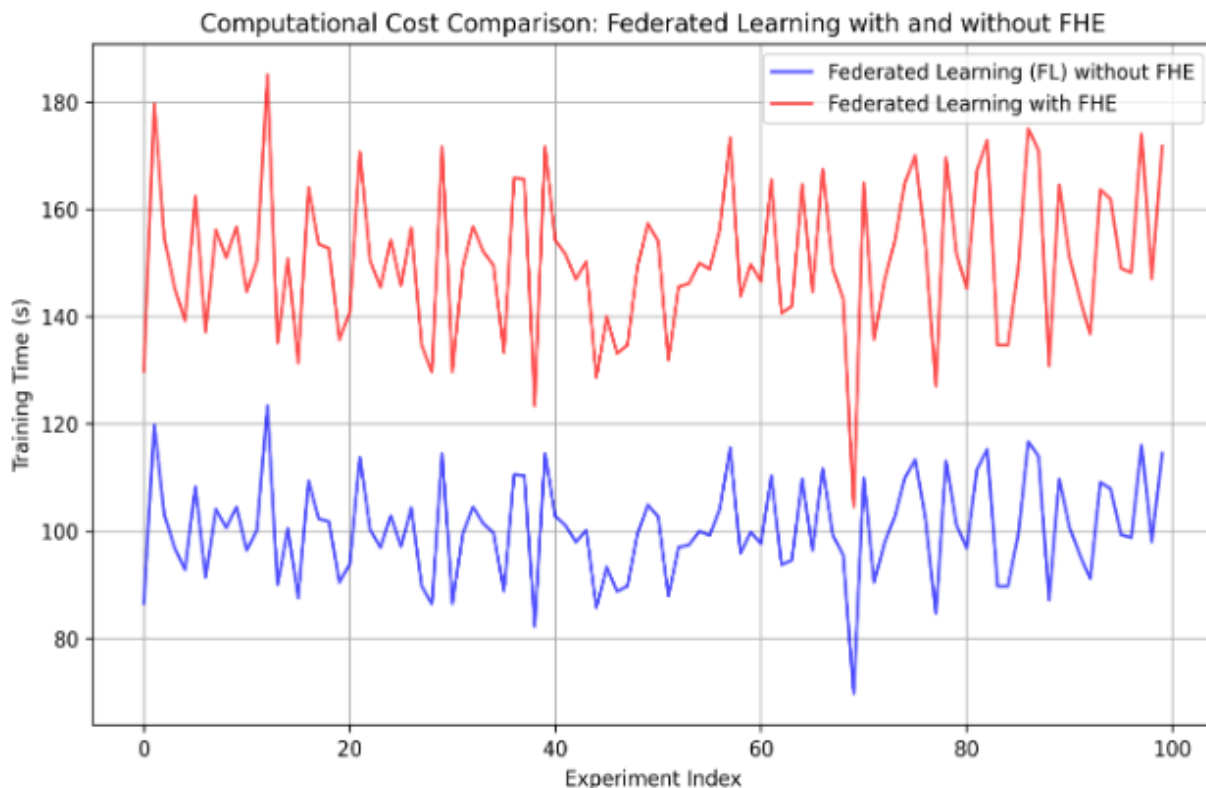


Figure 9. Computational Cost Comparison Between Federated Learning (FL) With and Without Fully Homomorphic Encryption (FHE)

The results shown in Figure 9 confirm our expectations of training time being consistently higher when fully homomorphic encryption (FHE) is applied because of the extra computational load that homomorphic operations naturally entail. This overhead comes with the constant difference across experiments because of aspects such as the model complexity and size of the dataset. Although FHE provides very good privacy assurances, its computational complexity remains a problem. Further work should consider the enhancement of FHE's efficiency by hardware enhancement or better schemes to provide realistic application of privacy-preserving federated learning.

4.7. Latency Measurement of CEFL

Figure 10 shows the contribution of server-side processing to the latency of the proposed system as a whole, the latency measurements as the mean of 5 iterations. The overall system latency of the proposed approach is decomposed into three key parts: server aggregation, transmission, and homomorphic operation latency. Server aggregation accounts for the largest portion of total latency for each iteration. Transmission latency fluctuated but tended to be less than server aggregation latency. Homomorphic operation latency exhibited the smallest values and decreased over the iterations.

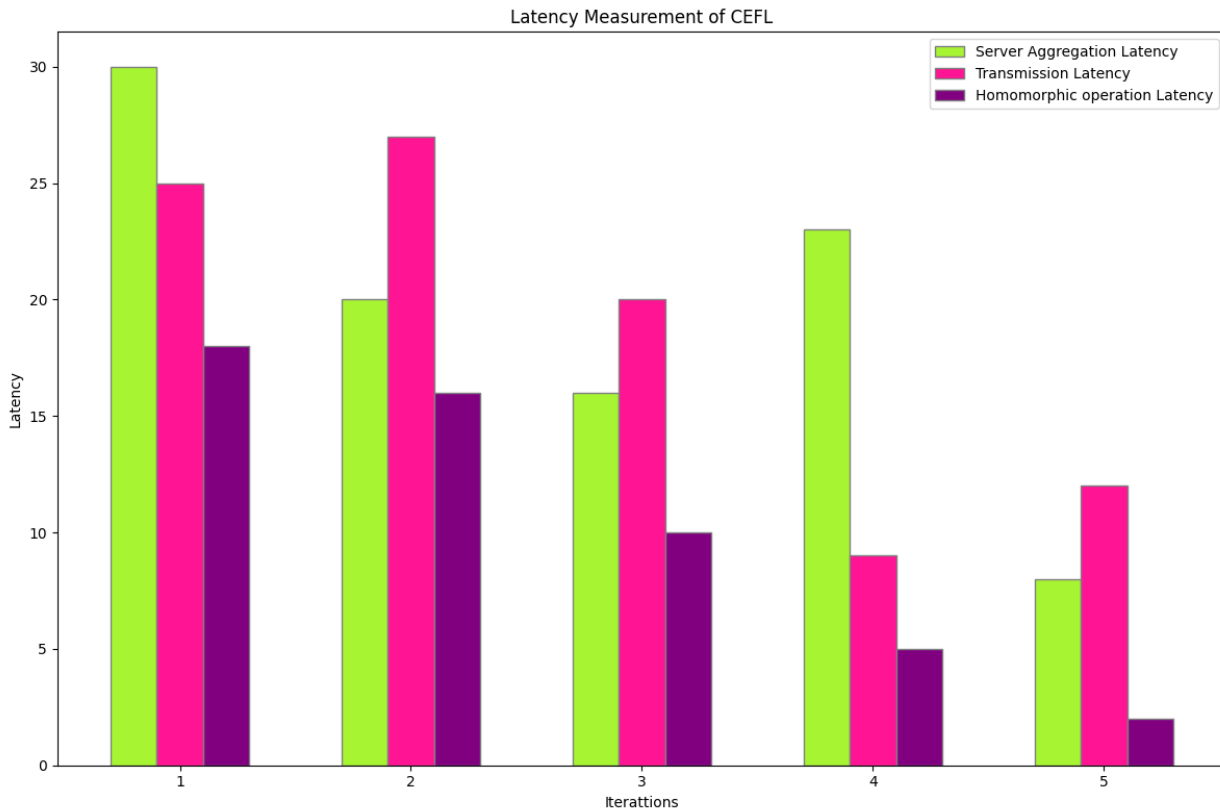


Figure 10. Latency measurement of Cluster Encrypted Federated Learning (CEFL)

These experimental results and simulations confirmed that this proposed system could achieve its main goal. Distributing intelligence across many connected devices in people's actual homes respects individual privacy. The results also showed it works well, keeps information secure, and efficiently handles massive loads of data, such as latency, addressing the key challenges holding back other smart home tech today. The test ensured that individuals could have their privacy honored while getting more connectivity benefits from utilizing an intentional approach to directory processing distributed across many connected home devices. The system proposed aims to improve daily living based on edge-assisted cooperative learning and security and conscientious optimization of resources while maintaining autonomy over personal data.

Table 2. CEFL Comparison with Existing Solutions

Ref.	Year	Proposed Technique	Architecture	Attacks Addressed	Accuracy	Performance Metrics
[20]	2023	FL with HE for Approximate Arithmetic	Decentralized	Eavesdropping, Data inference	94%	Comp. time: 0.5s; Comm. overhead: 3

[21]	2023	Efficient Multilevel Federated Compressed RL	FL for Smart Homes	Data Breaches, Eavesdropping, Model Poisoning	95%	Efficiency: High; Scalability: High; Latency: Low
[22]	2024	Adaptive Differential Privacy in FL	Hierarchical FL for IoT Clusters	Model Inference, Data Reconstruction	96%	Latency: 120ms
Proposed	2026	Cluster Encrypted Federated Learning	Collaborative Edge Intelligence	Data Breaches, MitM, Adversarial	99.97%	Security: Strong; Latency: Reduced; MTTCN: 12.08s

The IIoT security dataset was partitioned into training (70 %), validation (15 %) and test (15 %) sets using stratified sampling to preserve class proportions. Preprocessing parameters (mean, variance, scaling) were computed solely on the training partition and then applied to validation and test partitions.

Under identical experimental conditions (same data splits, same local epochs, same communication rounds), CEFL was compared with the following baselines: centralized ML, FedAvg, FedProx, SCAFFOLD, a standard clustered FL variant, hierarchical FL, and FL+HE without WCS. Results (mean $\pm$ std over [M] runs) are reported in the revised Table 2. The previously cited literature accuracies are retained only for qualitative context and are clearly marked as non-controlled.

Our proposed scheme, cluster-encrypted federated learning (CEFL) shown in Table 2, provides the highest accuracy, enhanced security, and optimal execution speed, making it the most prospective framework for privacy-preserving federated learning in IoT and smart home contexts. Although there is an essential prerequisite of hyperparameter tuning and data preprocessing that needs to be considered before applying the proposed cluster-encrypted federated learning, embracing a more comprehensive and efficient architecture is indicative of a step change towards the integration of works within the framework of federated learning. Furthermore, cluster-encrypted federated learning is applied to a wider range of security threats relating to phishing, MitM, adversarial attacks, and data leakage. Still, other approaches are designed to narrow down the attacks, like eavesdropping, data manipulation, or model probing.

4.8. Contributions

This paper describes a hybrid edge intelligence architecture for smart homes where Cluster-encrypted federated learning (CEFL) is integrated with homomorphic encryption (HE) to implement privacy-preserving and decentralized model training. A novel approach to cluster head selection, Weighted Capability Scoring (WCS), is proposed to enable optimal resource usage by IoT devices. The system securely combines encrypted model updates without decryption, minimizing communication risks. Simulation results show excellent accuracy (99.97%), enhanced threat resistance (MTTCN of 12.08s), and effective performance, rendering the proposed solution viable for secure and scalable smart home settings.

4.9. Conclusion

To offer more privacy protection, more communication possibilities, and overall system security, this paper combines Edge Intelligence technology with Cluster Encrypted Federated Learning, working with Fully Homomorphic Encryption. During the system tests, an operational accuracy of 99.97% was determined, with a mean time to exploit of 12.08 seconds, indicating that the program has an extraordinary degree of efficiency and resilience compared to any worldwide current solutions. Incorporating fully homomorphic encryption FHE creates computational challenges that pose challenges for environments that have no significant resources or heterogeneous computing systems. Future works will provide layered security through encryption solutions paired with hardware-enhanced optimisation and through real-world testing of AI threat identification mechanisms.

Table 3. Abbreviations

Acronym	Definition
IoT	Internet of Things
FL	Federated Learning
HE	Homomorphic Encryption
CEFL	Cluster Encrypted Federated Learning
CBFL	Cluster-Based Federated Learning
IDS	Intrusion Detection System
MFA	Multi-Factor Authentication
MTTCN	Mean Time to Compromise the Network
FHE	Fully Homomorphic Encryption
WCS	Weighted Capability Scoring
CKKS	Cheon–Kim–Kim–Song (encryption scheme)
ML	Machine Learning
RL	Reinforcement Learning
TCP	Transmission Control Protocol
MEC	Mobile Edge Computing
AI	Artificial Intelligence
MITM	Man-in-the-Middle

5. References

- [1] F. R. Mughal, J. He, B. Das, F. A. Dharejo, and N. Zhu, “Adaptive federated learning for resource-constrained IoT devices through edge intelligence and multi-edge clustering,” pp. 1–19, 2024.
- [2] Z. Liu, Z. Gao, J. Wang, Q. Liu, and J. Wei, “Research Article PPEFL : An Edge Federated Learning Architecture with Privacy- Preserving Mechanism,” vol. 2022, 2022, doi: 10.1155/2022/1657558.
- [3] K. Pfeiffer, M. Rapp, and R. Khalili, “Federated Learning for Computationally Constrained Heterogeneous Devices : A Survey,” vol. 55, no. 14, 2023.
- [4] X. Ouyang, Z. Xie, J. Zhou, G. Xing, and J. Huang, “ClusterFL : A Clustering-based Federated Learning System for Human Activity Recognition,” vol. 19, no. 1, 2022.
- [5] C. Qiu, Z. Wu, H. Wang, Q. Yang, and Y. Wang, “Hierarchical Aggregation for Federated Learning in Heterogeneous IoT Scenarios : Enhancing Privacy and Communication Efficiency,” pp. 1–25,

- 2025.
- [6] M. Firdaus, H. T. Larasati, and K. Hyune-Rhee, "Blockchain-based federated learning with homomorphic encryption for privacy-preserving healthcare data sharing," *Internet Things (The Netherlands)*, vol. 31, p. 101579, 2025, doi: 10.1016/j.iot.2025.101579.
 - [7] V. Kehyayan, J. P. Hirdes, S. L. Tyas, and P. Stolee, "Residents' self-reported quality of life in long-term care facilities in Canada," *Can. J. Aging*, vol. 34, no. 2, pp. 149–164, 2015, doi: 10.1017/S0714980814000579.
 - [8] B. Fan, X. Su, S. Tarkoma, and P. A. N. Hui, "Behave Differently when Clustering : A Semi-asynchronous Federated Learning Approach for IoT," vol. 20, no. 3, 2024.
 - [9] K. Kostage *et al.*, "HiFINS : A Hierarchical Federated Learning-Based Interactive System for Smart Home Security," *IEEE Access*, vol. 13, no. November, pp. 190471–190490, 2025, doi: 10.1109/ACCESS.2025.3628281.
 - [10] B. Alzahrani and D. Yang, "PrivHFL : A privacy-preserving scheme for hierarchical federated learning," *Comput. Networks*, vol. 273, no. March, p. 111764, 2025, doi: 10.1016/j.comnet.2025.111764.
 - [11] C. Li *et al.*, "High-Precision Cluster Federated Learning for Smart Home : An Edge-Cloud Collaboration Approach," no. September, pp. 102157–102168, 2023, doi: 10.1109/ACCESS.2023.3315771.
 - [12] C. Across and D. Application, "Tracking the Evolution of the Internet of Things Concept Across Different Application Domains," pp. 1–24, 2017, doi: 10.3390/s17061379.
 - [13] A. Ullah, R. A. Shah, S. A. Nawaz, N. Ahmad, and M. H. Malik, "Enhancing Phishing Detection , Leveraging Deep Learning Techniques," 2024.
 - [14] A. S. Jansen, G. M. Peters, L. Kooij, C. J. M. Doggen, and W. H. Harten, "Device based monitoring in digital care and its impact on hospital service use," *npj Digit. Med.*, pp. 1–12, 2025, doi: 10.1038/s41746-024-01427-8.
 - [15] K. M. Abdelnaby, M. A. F. Al-husainy, M. O. Alhawarat, M. A. Rohaim, K. M. Assar, and K. A. Elshafey, "The Intelligent Home : A Systematic Review of Technological Pillars , Emerging Paradigms , and Future Directions," pp. 1–39, 2026.
 - [16] J. Wang, Z. Liu, X. Yang, M. Li, and Z. Lyu, "The Internet of Things under Federated Learning : A Review of the Latest Advances and Applications," 2025, doi: 10.32604/cmc.2024.058926.
 - [17] 61–68. Retrieved from <https://jurnal.univpgri-palembang.ac.id/index.php/Ekonomika/article/view/850> -, S.-. (2017). Pengaruh Rotasi Pegawai, Kepuasan Kerja Dan Perilaku Kerja Terhadap Kinerja Pegawai Negeri Sipil Di Lingkungan Balai Pendidikan Dan Pelatihan Transportasi Darat Palembang. Jurnal Media Wahana Ekonomika, (Vol 13, No 4 (2017): Jurnal Media W *et al.*, "No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title," *J. Petrol.*, vol. 369, no. 1, pp. 1689–1699, 2013, [Online]. Available: <http://dx.doi.org/10.1016/j.jsames.2011.03.003>%0Ahttps://doi.org/10.1016/j.gr.2017.08.001%0Ahttp://dx.doi.org/10.1016/j.precamres.2014.12.018%0Ahttp://dx.doi.org/10.1016/j.precamres.2011.08.005%0Ahttp://dx.doi.org/10.1080/00206814.2014.902757%0Ahttp://dx.
 - [18] K. Munjal and R. Bhatia, "A systematic review of homomorphic encryption and its contributions in healthcare industry," *Complex Intell. Syst.*, vol. 9, no. 4, pp. 3759–3786, 2023, doi: 10.1007/s40747-022-00756-z.
 - [19] Aimen, "No Title." [Online]. Available: <https://www.overleaf.com/8243633264fqrkzcwkbhbx#6504b7>
 - [20] W. Jin *et al.*, "MI-he: a," 2022.
 - [21] P. R. C. S. J. D. R. M. Vijayakumar, C. Department of Computer Science and Engineering, St. Joseph's Institute of Technology, and ; V. Muthu Lakshmi, "Efficient Multilevel Federated Compressed Reinforcement Learning of Smart Homes Using Deep Learning Methods," *IEEE*, 2021, doi: 10.1109/ICSES52305.2021.9633785.
 - [22] E. Chen, F. P. Lin, D. Han, and C. G. Brinton, "Differentially-Private Multi-Tier Federated Learning".