LAHENet: A Lightweight Additive Homomorphic Edge Neural Network Framework for Industrial IoT

Mowei Gong, Zhe Li, Xuepeng Lu, Bei Gong, *Member, IEEE*, Haotian Zhu, Weizhi Meng, *Senior Member, IEEE*

***Abstract*—Edge nodes in the Industrial Internet of Things (IIoT) often face a fundamental trade-off between limited computational resources and stringent real-time inference requirements. Moreover, sensitive data they generated are exposed to significant privacy and security threats during transmission and computation. To address these challenges, this paper proposes a lightweight additive homomorphic edge neural network framework called LAHENet. This framework achieves millisecond-level inference latency in real-world industrial environments through a combination of a dual-metric feature selection strategy, an efficient additive homomorphic signcryption protocol, and a lightweight linear computation layer with adaptive layer collapsing. It ensures end-to-end confidentiality, unforgeability, forward security, and verifiable computation correctness. Experimental results show that LAHENet maintains a constant communication overhead at a few kilobytes per inference while preserving high model accuracy. It significantly enhances inference efficiency and reduces bandwidth consumption in edge environments, offering a practical private inference solution for large-scale IIoT deployments.**

***Index Terms*—Private inference, additive homomorphic encryption, lightweight, Industrial Internet of Things (IIoT).**

I. Introduction

The advance of Industry 5.0 is driving a paradigm shift in key sectors such as manufacturing, energy, and transportation from technology-centric automation to human-centric collaboration. This paradigm integrates edge computing and artificial intelligence at the production site to meet the demands for flexible, small-batch customization, balancing efficiency, resilience, and sustainability [1]. The Industrial Internet of Things (IIoT) is a cornerstone of this transition, providing high-speed sensing and connectivity. The dense deployment of billions of IIoT devices across industrial sites generates massive, high-velocity, and diverse data streams. To leverage these data for critical applications such as predictive maintenance and real-time scheduling, the computing paradigm is shifting towards the network edge. This shift enables millisecond-level analytics near the data source, mitigating decision risks from transmission delays and bandwidth bottlenecks [2].

Industry forecasts project that the economic impact of integrating edge intelligence with IIoT will exceed trillions of dollars within the next decade [1]. However, the deep interconnection, real-time intelligence, and large-scale deployment inherent to Industry 5.0 also introduce complex data security and privacy risks [3]. IIoT data often contain sensitive production parameters and critical operational information. Furthermore, numerous heterogeneous edge nodes are deployed in physically accessible locations or connected to legacy systems with inadequate security, creating a vast attack surface. Concurrently, these edge nodes must perform millisecond-level intelligent inference while preserving data confidentiality and integrity within resource-constrained environments. The high overhead of traditional encryption and key-based access mechanisms places additional pressure on the already limited resources and stringent low-latency requirements of these devices [4]. Consequently, ensuring end-to-end privacy and integrity while meeting the demands for real-time inference and lightweight deployment has become a critical technical challenge.

Traditional solutions to these challenges involve transmitting sensor data to the cloud for centralized analysis and processing. Although this can leverage abundant computing resources, network interruptions or congestion may disrupt control loops, and cross-domain transmission and third-party storage expose data to leakage risks. To alleviate the security threats posed by transmission and storage, some methods combine blockchain with threat intelligence or data-exchange mechanisms to provide on-chain traceability through multi-party consensus [5]. However, these solutions still require additional isolation for the inference stage within individual nodes. Edge-based local inference schemes avoid these transmission-related issues but have their own significant limitations. Privacy-preserving machine learning (PPML) methods, such as those based on fully homomorphic encryption (FHE), permit arbitrary computations on ciphertext but incur substantial computational overhead and high ciphertext expansion rates [6], [7]. Even with lightweight variants tailored for IIoT [8], [9], their latency is insufficient for millisecond control windows, and the overhead from interaction rounds remains substantial. Approaches combining re-encryption and federated learning (FL) [10], [11], [12] reduce raw data exposure but suffer from accuracy loss and slow convergence on data

(Corresponding authors: Bei Gong and Weizhi Meng. Co-first authors: Mowei Gong and Zhe Li)

Mowei Gong, Zhe Li and Haotian Zhu are with the College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: gongmowei@emails.bjut.edu.cn, leezhe627@emails.bjut.edu.cn, zhuhaotian@emails.bjut.edu.cn).

Xuepeng Lu is with Beijing Luo'an Technology Co., Ltd., Beijing 100085, China (e-mail: luxp@icssla.com).

Bei Gong is with the Beijing Key Laboratory of Trusted Computing, College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: gongbei@bjut.edu.cn).

Weizhi Meng is with the School of Computing and Communications, Lancaster University, UK (e-mail: weizhi.meng@ieee.org).

Manuscript received XXXXX; revised XXXXXX.

that are not independently and identically distributed (Non-IID), a data type prevalent in IIoT. Furthermore, their architecture lacks effective defences against malicious or compromised central servers. Consequently, a unified framework capable of simultaneously balancing millisecond-level inference efficiency with end-to-end privacy protection for industrial edge applications has yet to emerge.

To address this gap, we propose a lightweight additive homomorphic edge neural network framework (LAHENet) for efficient, secure, and private inference in industrial edge scenarios. It establishes a collaborative secure-computing model between resource-constrained nodes and more powerful but untrusted nodes. By integrating efficient additive homomorphic signcryption protocols with lightweight optimizations of a private inference linear computing layer, LAHENet provides end-to-end protection for the full lifecycle of industrial data while satisfying the real-time performance demands of industrial applications. The main contributions of this study are as follows.

- **Lightweight.** In the offline phase, we employ a hybrid feature selection method to perform data dimensionality reduction. In the online phase, the framework adaptively collapses network layers according to node resources and uses precomputation accelerators to minimize computational and communication overhead.
- **Efficient.** We propose a novel and efficient additive homomorphic signcryption protocol that automatically fuses encryption and signing. It leverages physical unclonable functions (PUFs) for on-demand key reconfiguration, eliminating complex key management burdens and enabling direct inference on ciphertexts, which significantly reduces the overhead.
- **Trustworthy Encrypted Inference.** We design an inference mechanism that operates entirely based on encrypted data. It enables zero-interaction verification for real-time integrity checks and provides forward security to guarantee the long-term safety of historical data, ensuring that the results are both immediately trustworthy and retrospectively secure.
- **Provable Security.** We show that the LAHENet guarantees confidentiality (IND-CCA2), unforgeability (EUF-CMA), forward security, and verifiable computation correctness by providing a formal security proof under the random oracle model.

The remainder of this paper is structured as follows. Section II reviews the related prior work on private inference and security mechanisms. Section III presents the cryptographic preliminaries and describes the system model, threat model, security goals, and assumptions of the LAHENet framework. We present the detailed construction of the LAHENet framework in Section IV. The security and performance of LAHENet are analysed and discussed in Sections V and VI, respectively. Finally, Section VII summarizes this work.

II. Related Work

A. Homomorphic Encryption in Private Inference

Homomorphic encryption (HE) is a key technology for private inference, enabling direct computation on the basis of ciphertexts. CryptoNets [6] first demonstrated the feasibility of HE via polynomial approximation but suffered from high computational overhead and poor accuracy compared with plaintext methods. Subsequent research has therefore focused on the codesign of cryptographic protocols and models, as well as computation optimizations.

In terms of codesign, Chou et al. [13] used network pruning to reduce expensive ciphertext multiplication but at the cost of model accuracy. Chen et al. [14] developed a CKKS-based Transformer workflow, where large-scale bootstrapping remains a significant performance bottleneck. For computational optimization, Lou et al. [15] proposed a framework named SHE for precise ReLU implementation using Boolean operations, although bit-level computation is inefficient for large-scale numerical tasks. More recently, Harb and Celiktas [16] used a lightweight neural network to replace the activation approximation, which reduced the multiplicative depth but increased the overall system complexity.

Moreover, to overcome performance bottlenecks in linear operations, other works have introduced specific optimizations. Kim et al. [17] proposed a bootstrapping-aware constant-time convolution with polynomial activation, although it is difficult to adapt to attention modules. Wu et al. [18] extended vector packing to the Paillier cryptosystem to compress communication, but this requires a trade-off between bandwidth and flexibility. Liu et al. [19] achieved lower latency for medical image inference through scenario-specific quantization, yet their method is highly dependent on the data distribution. Recent studies [20], [21] have successfully halved the relinearization cost of attention layers using techniques such as diagonal encoding and ciphertext compression. Despite this progress, both approaches still face challenges with key size expansion and noise accumulation. Current HE-based private inference methods have advanced in terms of packing strategies, activation approximation, and network codesign, but they predominantly rely on complex RLWE-based cryptosystems. High computational overhead, large ciphertext sizes, and complex key management impose severe demands on node resources, creating a core barrier to large-scale deployment.

B. Secure Interactivity in Private Inference

Interactive secure inference distributes cryptographic costs across multiple parties, performing heavy computations offline to ensure low-latency online inference. MiniONN [22] is an early example of the use of secret sharing and garbled circuits (GC) for efficient nonlinear activation. However, its high round count and large ciphertext transmissions lead to substantial bandwidth consumption.

Subsequent work has focused on reducing this overhead. Riazi et al. [23] used binary neural networks to achieve a constant number of rounds, but this approach sacrificed model accuracy. Similarly, Boemer et al. [24] employed a client-aided architecture to enable high-throughput inference on deep networks. Despite its efficiency, this design inherently exposes intermediate pre-activation values to the client, posing a risk of model extraction. Others have designed single-round ReLUs using function secret sharing (FSS) [25] or reduced ReLU counts via neural architecture search (NAS) [26], although

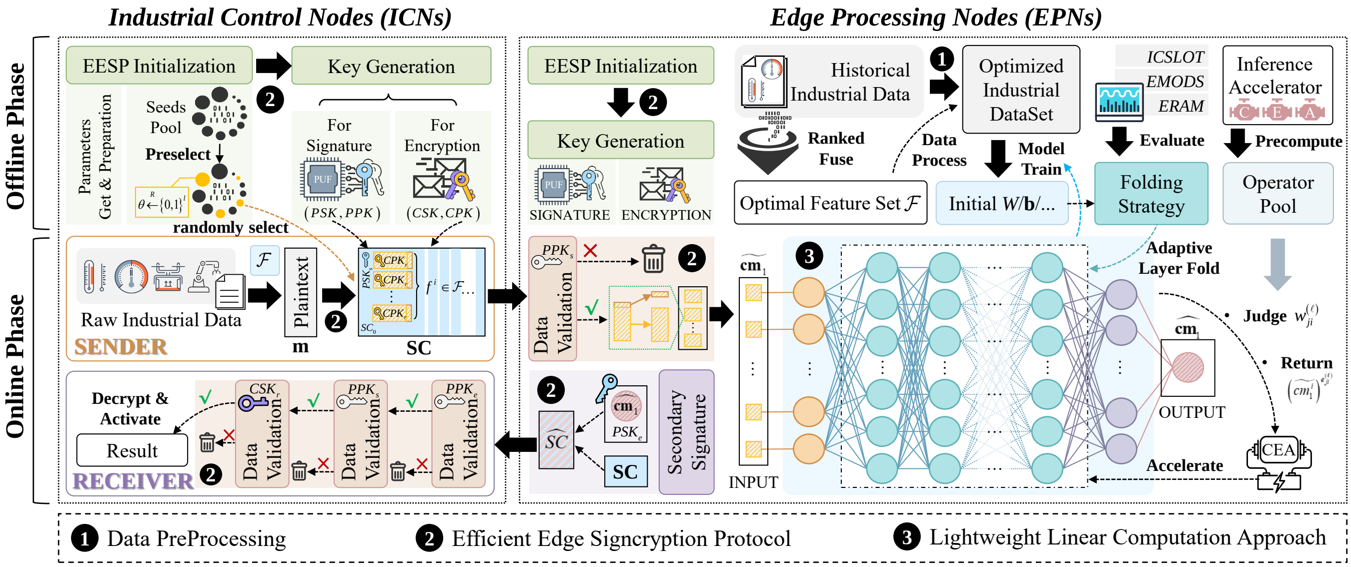


Fig. 1. The workflow of LAHENet.

these methods require a trusted setup or are computationally expensive, respectively. Hu et al. [27] combined HE with additive secret sharing to cut communication, yet ciphertext expansion and conversion delays persisted.

To avoid latency from switching between primitives, Yang et al. [28] unified all operations under a single cryptographic primitive. However, it increased computational and storage costs. This interactive inference has also been applied to Transformers [29], [30]. Yet, efficient attention mechanisms come at the cost of high overhead and approximate model modifications. Recently, Fu et al. [31] proposed a scheme that uses encoding-based conversion to speed up nonlinear functions, although it increases system complexity. Although interactive secure inference has improved in terms of reducing rounds and overhead, the overhead from ciphertext-plaintext conversions and communication bursts in nonlinear protocols limits its application in deep models and complex network environments.

C. Hybrid Optimizations for Private Inference

Hybrid optimization strategies aim to balance the efficiency and security of private inference. Juvekar et al. [32] proposed GAZELLE with lower latency by mixing cryptographic protocols, but it causes high communication overhead. To reduce this cost, later frameworks [33], [34] moved HE computations offline or used automated parameter tuning, but both remain multiround protocols sensitive to network latency. Singh et al. [35] use a cost model to adapt tensor packing and computation strategies but fail to address the challenge of nonlinear computation.

Another approach combines hardware trusted execution environments (TEEs) with model partitioning. Rajasekar et al. [36] explored partitioning but exposed intermediate features in plaintext. TEE-based schemes offer stronger security. Liu et al. [37] reduced the TEE load through model splitting and mirror networks, although splitting points can leak partial results. Other solutions fuse FHE with TEEs [38] or use table lookups [39] to increase performance, but frequent memory boundary interactions and limited TEE memory restrict their use to less complex models. Hybrid schemes balance efficiency and security but remain constrained by protocol complexity and hardware limitations.

D. Lightweight Security for the IIoT Edge

A primary challenge for private inference on resource-constrained IIoT edges is to simultaneously guarantee data security and computational integrity. To meet this need within tight resource constraints, initial research focused on lightweight cryptography. Ullah et al. [40], [41] proposed hyperelliptic curve-based signcryption schemes, but limited group operation performance and certificate management costs hinder scalability. Kim et al. [42] extended LiSP for lightweight signcryption but ignored recovery after key leakage. Rao et al. [43] introduced a certificateless aggregated signcryption method that integrates encryption and authentication, which risks single-point failure. To address key storage, PUFs were introduced to establish a hardware root of trust. Barbareschi et al. [44] proposed a pseudo-PUF that reduces costs but suffers from environmental reliability issues. Efficient PUF-based anonymous authentication [45] and key exchange [46] have been proposed, but both are difficult to deploy at scale. A recent cross-layer PUF protocol [47] enhances security, but its design complicates deployment.

These studies primarily secure communication channels and endpoint identities. However, they do not solve privacy leakage during computation on untrusted edge servers. Thus, building a truly end-to-end, lightweight private inference framework for resource-constrained IIoT environments remains a critical research gap.

III. Preliminaries

A. System Model

LAHENet involves industrial control nodes (ICNs) located at the perception layer and edge processing nodes (EPNs) at the edge of the IIoT. The ICN is responsible for real-time sensing, acquisition, and signcryption of raw industrial data. The EPN performs efficient linear inference in the ciphertext domain with greater computational and memory resources.

We describe the workflow of LAHENet in Fig.1. This process operates under a public-model assumption, where linear model parameters are treated as shared system configurations available to both parties. Crucially, this design ensures that while the model configuration is shared, the sensitive plaintext data remains inaccessible to the EPN. Instead, the receiver locally validates the linear computation result against these shared configurations to ensure integrity, and subsequently applies the nonlinear activation.

LAHENet works in two phases, namely, the offline precomputation phase and the online secure inference phase.

- **Offline Precomputation Phase:** To reduce the computational and communication overhead caused by high-dimensional feature redundancy in IIoT environments, LAHENet adopts a two-stage feature evaluation strategy. Specifically, the EPN selects an optimal feature subset using both analysis of variance (ANOVA) and mutual information. A ranking-weighted fusion strategy is applied to identify features that offer the best combination of discriminative power and information content. During this phase, the EPN and ICN also perform system initialization, generate keys for additive homomorphic encryption, and assess resources for adaptive layer folding.
- **Online Secure Inference Phase:** This online phase is jointly executed by the ICN and EPN. First, the ICN extracts features from the optimal subset and signcrypts them with a random seed and the receiver's public key. Then, the ICN sends the ciphertext to the EPN. After verifying authenticity and integrity, the EPN performs private inference directly on the encrypted data using a linear neural network. It is important to note that the EPN operates exclusively in the ciphertext domain and never accesses the plaintext. To enhance efficiency, the EPN applies an adaptive layer-folding strategy, fusing weights and biases or performing layerwise homomorphic accumulation based on network structure and resource limits. Finally, the receiver verifies, decrypts, and applies a nonlinear activation to obtain the final real-time private inference output for IIoT scenarios.

B. Threat Model

The LAHENet framework aims to efficiently and securely enable private inference in IIoT edge scenarios. An ICN at the source signcrypts the raw industrial data immediately after generation and sends it to an EPN, which is allowed to perform only additive homomorphic linear inference. An ICN, as the legal receiver, verifies and decrypts the computation result. Afterward, the final plaintext is applied to the nonlinear activation function. We operate under a public-model assumption where the linear model parameters are treated as shared system configurations. Consequently, the adversary gains no advantage in compromising data confidentiality or verifiable correctness by accessing these parameters.

Owing to open communication links and limited node resources in IIoT, we model potential threats as a joint adversary $\mathcal{A} = (\mathcal{A}_{net}, \mathcal{A}_{node})$, where $\mathcal{A}_{net}$ and $\mathcal{A}_{node}$ can share internal states and act in coordination.

- **Communication link adversary** $\mathcal{A}_{net}$: It controls the untrusted network between ICNs and EPNs. $\mathcal{A}_{net}$ covers typical link attacks, including eavesdropping, man-in-the-middle attacks, and replay attacks.
- **Node adversary** $\mathcal{A}_{node}$: It can adaptively corrupt all EPNs and up to $t-1$ ICNs. ICNs expose the long-term private key, and the EPN outputs forges linear results or injects malicious ciphertexts after corruption.

With these adversary capabilities, we consider the following four types of attacks. However, any attack capable of physically replicating or probing the internal structure of a PUF is not discussed in this paper. The unclonability and unpredictability of PUFs are assumed in Section III-E.

1) *Link attacks*. $\mathcal{A}_{net}$ listens to and modifies messages in transit, attempting to obtain the plaintext or destroy message integrity.
2) *Single-node snooping attacks*. $\mathcal{A}_{node}$ corrupts an EPN or ICN, attempting to reconstruct the plaintext from the partial ciphertext.
3) *Multinode collusion attacks*. Multiple corrupted nodes share their ciphertexts and internal states, attempting to obtain complete private data or conceal evidence of tampering.
4) *Malicious computation and forgery attacks*. A corrupted EPN or ICN forges ciphertexts, returns incorrect linear results, or alters historical messages after the key is exposed.

Currently, LAHENet addresses privacy breaches in the transmission and edge computing phases of industrial data, whereas attacks in adversarial machine learning, and PUF erosion are left for future work.

C. Security Goals

A private inference framework for the IIoT must withstand a range of attacks, as mentioned in Section III-B, to provide formal security guarantees. Therefore, LAHENet should achieve the following security goals.

- **Confidentiality**. In any session, no unauthorized entity can learn any information from signcrypted information or homomorphic computations. Only the intended receiver can decrypt.
- **Unforgeability**. Receivers must be able to verify the origin and integrity of the ciphertext. An adversary cannot forge new valid signcryptions or undetectably modify any ciphertext, even with access to historical data.
- **Forward Security**. Even if the long-term key of an ICN is exposed, past signed data and inference results remain protected from decryption or tampering.
- **Verifiable Computational Correctness**. Any computation performed by a malicious EPN that deviates from the public model must be detected and rejected by the recipient.

D. Cryptographic Preliminaries

1) Additive Homomorphic Encryption (AHE)

An encryption scheme is additively homomorphic if the encryption satisfies $\mathrm{Dec}(c_1 \oplus c_2) = m_1 + m_2$ and $\mathrm{Dec}(k \odot c_2) = k \cdot m_2$, where $c_1 = \mathrm{Enc}(m_1)$, $c_2 = \mathrm{Enc}(m_2)$, k is a known integer and $\oplus$, $\odot$ denote addition and scalar multiplication in the ciphertext domain.

2) Paillier Cryptosystem

The Paillier cryptosystem [48] is a public-key encryption scheme over $\mathbb{Z}_{n^2}^{\times}$ with modulus $n = pq$, where p and q are large primes. A message $m \in \mathbb{Z}_n$ is encrypted as $c = g^m r^n \bmod n^2$, where $g \in \mathbb{Z}_{n^2}^{\times}$ and $r \in \mathbb{Z}_n^{\times}$ is randomly chosen. Decryption recovers $m = \mathrm{L}(c^{\lambda} \bmod n^2) \cdot \mu \bmod n$, where $\mathrm{L}(x) = \frac{x-1}{n}$, $\lambda = \mathrm{lcm}(p-1, q-1)$, and $\mu = \mathrm{L}(g^{\lambda} \bmod n^2)^{-1} \bmod n$.

It supports homomorphic addition and plaintext-ciphertext multiplication, which instantiates the AHE operations by mapping $\oplus$ to multiplication in $\mathbb{Z}_{n^2}^{\times}$ and $\odot$ to exponentiation by a known integer. Equivalently, for any $m_1, m_2 \in \mathbb{Z}_n$ and known integer k, $\mathrm{Enc}(m_1) \oplus \mathrm{Enc}(m_2) \equiv \mathrm{Enc}(m_1 + m_2) (\mathrm{mod}\ n^2)$, $k \odot \mathrm{Enc}(m_1) \equiv \mathrm{Enc}(km_1) (\mathrm{mod}\ n^2)$, where all products and exponentiations are taken in $\mathbb{Z}_{n^2}^{\times}$.

3) PUFs

A physical unclonable function can be modelled as $\mathrm{PUF} : C \to R$, where C is the challenge space and R is the response space. Given a challenge $C_i \in C$, the device's PUF circuitry outputs $R_i = \mathrm{PUF}(C_i)$, which is stable under normal conditions and unique to the specific hardware due to manufacturing variations.

E. Security Assumptions

To ensure the security of LAHENet, we base the subsequent security analysis on the following two security assumptions. Let κ be the security parameter and $\mathbb{Z}_n = \{0, \ldots, n-1\}$. The signcryption in LAHENet uses the Paillier additive homomorphic cryptosystem, with all operations modulo n^2, where $n = pq$ is the product of two equal-length primes.

Assumption 1 *(Decisional Composite Residuosity (DCR))*: Let $n = pq$ and $g = n + 1$ with p, q equal-length odd primes. Choose $y \in \mathbb{Z}_{n^2}^{\times}$, $m \in \mathbb{Z}_n$ uniformly at random, and then pick $b \in \{0, 1\}$ uniformly at random to construct $X = (y^n)^{1-b} \cdot (g^m y^n)^b \bmod n^2$. If $b = 0$, $X \in \mathbb{R}_n$, where $\mathbb{R}_n = \left\{ z^n \bmod n^2 \middle| z \in \mathbb{Z}_{n^2}^{\times} \right\}$. Otherwise, $X \in g^m \mathbb{R}_n$. For any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, $|\Pr[\mathcal{A}(n, g, X) = 1 | b = 0] - \Pr[\mathcal{A}(n, g, X) = 1 | b = 1]| \leq \mathrm{negl}(\kappa)$.

Assumption 2 *(PUFs Security)*: Each node is equipped with a unique PUF. For any PPT adversary $\mathcal{A}$, the probability of producing a functionally equivalent PUF or correctly predicting the response to an unseen challenge is negligible.

IV. LAHENet Framework

A. Design Overview

The LAHENet framework addresses the limited computing and communication capacity of IIoT edge devices through a staged secure inference design that balances real-time performance and data security. In the offline preprocessing stage, ANOVA and mutual information identify features most relevant to the inference task, reducing encryption and transmission costs (in Section IV-B). In the online stage, LAHENet first employs a PUF-bound Paillier-based additive homomorphic signcryption protocol to protect data privacy (in Section IV-C). It then aggregates ciphertext in a single pass using additive homomorphism, with adaptive layer folding and an inference accelerator enabling efficient ciphertext-domain linear computation, thereby enhancing edge processing capability and overall system responsiveness (in Section IV-D).

B. Preprocessing for LAHENet

In IIoT edge scenarios, ICNs can acquire raw feature vectors with more than 100 dimensions in a single collection cycle. However, only a small subset is strongly correlated with the inference target, and processing numerous redundant features wastes computational resources. To address this issue, we proposed a static feature selection mechanism based on a two-layer evaluation in the LAHENet framework. This mechanism operates during the offline training phase to identify a subset of highly discriminative features. During the offline training phase, it selects a subset of highly discriminative features, enabling targeted dimensionality reduction before encryption and reducing the online computational burden.

We first calculated the linear correlation between each feature xf_i and the target variable $\mathcal{T}$ via one-way ANOVA with Eq. (1).

$$F(xf_i) = \frac{\sum_{\varepsilon=1}^{K} n_{\varepsilon} \left(\overline{xf}_{i,\varepsilon} - \overline{xf}_i \right)^2 / (K-1)}{\sum_{\varepsilon=1}^{K} \sum_{d=1}^{n_{\varepsilon}} \left(xf_{i,d,\varepsilon} - \overline{xf}_{i,\varepsilon} \right)^2 / (N-K)} \tag{1}$$

where K represents the number of categories in $\mathcal{T}$, n_{ε} is the number of samples in the ε-th category, $\overline{xf}_{i,\varepsilon}$ is the mean of xf_i within that category, $\overline{xf}_i$ is the global mean of the whole sample, and N is the total number of samples. We record the first n features of the $F(xf_i)$ result into the linear candidate set $\mathcal{F}_1$.

Next, we compute the nonlinear correlation features of the data with Eq. (2).

$$\tilde{M}(xf_i; \mathcal{T}) = \frac{M(xf_i; \mathcal{T})}{\min[H(xf_i), H(\mathcal{T})]} \tag{2}$$

where $H(xf_i)$ and $H(\mathcal{T})$ represent the information entropy of xf_i and $\mathcal{T}$, respectively. We record the first m features of the $\tilde{M}(xf_i; \mathcal{T})$ result into the nonlinear candidate set $\mathcal{F}_2$.

The final optimal feature set $\mathcal{F}$ is then defined with Eq. (3).

$$\mathcal{F} = \mathrm{TOP}\left(\alpha \cdot R_F(xf_i) + (1 - \alpha) \cdot R_{MI}(xf_i) \right) \tag{3}$$

where $xf_i \in \mathcal{F}_1 \cup \mathcal{F}_2$, α is the balancing weight, and $R_F(xf_i)$ and $R_{MI}(xf_i)$ represent the percentage rankings of xf_i in $F(xf_i)$ and $\tilde{M}(xf_i; \mathcal{T})$, respectively. We empirically set $\alpha = 0.35$ via 5-fold cross-validation over the validation set, balancing linear separability and mutual information. A simple decision rule then fixes the subset size dimension on a held-out validation split under a predefined accuracy budget δ. The smallest dimension that satisfies Eq. (4) is selected.

$$k_{dim}^{\star} = \min\{k_{dim} : Acc(k_{dim}) \geq Acc_{full} - \delta\} \tag{4}$$

where δ is the predefined accuracy budget, k_{dim} is a candidate subset size, and $k_{dim}^{\star}$ is the selected size. The Acc_{full} denotes the validation metric computed with all features,

TABLE I
NOTATIONS

Notation	Description
$\mathbb{Z}_n$	Residue ring of integers modulo n
$\mathbb{Z}_n^{\times}$ or $\mathbb{Z}_{n^2}^{\times}$	Multiplicative group of integers modulo n or n^2
GP	System public parameters
(CPK, CSK)	Communication key pair
(PPK, PSK)	Signature key pair
(C_i, R_i)	The i-th challenge–response pair
Carm (n)	The Carmichael function
θ	Signcryption session seed
ξ	Multiplicative mask in signcryption
ρ	Ciphertext re-randomization factor
τ	Message integrity tag

Notes: We use $(\cdot)_s$ to indicate parameters associated with the sender, $(\cdot)_r$ indicates parameters related to the receiver, and $(\cdot)_e$ indicates parameters belonging to EPN.

$Acc(k_{dim})$ denotes the metric computed with the top k_{dim} features induced by Eq. (3). With the deployed subset $\mathcal{F}_{final} = \mathrm{TOP}_{k_{dim}^{\star}}(\cdot)$, it maintains selection reproducibility while bounding changes in accuracy, reducing the encryption workload and ciphertext traffic.

In the online inference stage, the ICN encrypts only the features within this selected subset $\mathcal{F}_{final}$. This approach directly reduces the computational load for encryption and mitigates the communication overhead caused by ciphertext expansion. Based on application-specific management policies, these features, which are outside the set, are either stored locally or discarded.

C. Efficient Edge Signcryption Protocol

The secure online inference of LAHENet involves two main tasks: efficient edge signcryption and lightweight private inference. This section details the first task, and an efficient additive homomorphic signcryption protocol designed for the IIoT edge is proposed. The detailed flows of the system initialization and key generation are shown in Fig. 2, the detailed flows of the signcryption, process and forward, and unsigncryption phases are shown in Fig. 3. The main notations and their descriptions are given in Table I.

(1) *System Initialization Phase*

- $Setup(\kappa) \to GP$

Given the security parameter κ, EPN selects four collision-resistant one-way hash functions: $H_0^{(n)} : \{0,1\}^* \to \mathbb{Z}_n^{\times}$, $H_1 : \{0,1\}^* \to \{0,1\}^l$, $H_2^{(\eta)} : \{0,1\}^* \to \mathbb{Z}_\eta$, $H_3 : \{0,1\}^* \to \{0,1\}^{l_{tag}}$, where l is the security length and l_{tag} is the tag length of the data. Subsequently, EPN publishes the system parameters $GP = \left\{H_0^{(\cdot)}, H_1, H_2^{(\cdot)}, H_3\right\}$ to the ICN.

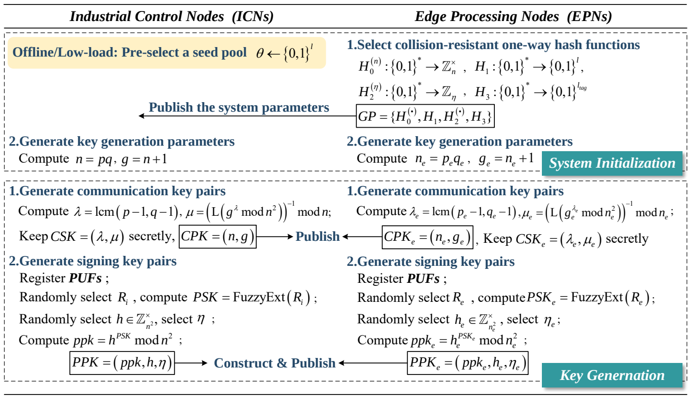

Industrial Control Nodes (ICNs)
Edge Processing Nodes (EPNs)
Offline/Low-load: Pre-select a seed pool $\theta \leftarrow \{0,1\}^l$
1.Select collision-resistant one-way hash functions
$H_0^{(n)} : \{0,1\}^* \to \mathbb{Z}_n^{\times}$, $H_1 : \{0,1\}^* \to \{0,1\}^l$,
$H_2^{(\eta)} : \{0,1\}^* \to \mathbb{Z}_\eta$, $H_3 : \{0,1\}^* \to \{0,1\}^{l_{tag}}$
Publish the system parameters
$GP = \{H_0^{(\cdot)}, H_1, H_2^{(\cdot)}, H_3\}$
2.Generate key generation parameters
Compute $n = pq$, $g = n+1$
2.Generate key generation parameters
Compute $n_e = p_e q_e$, $g_e = n_e + 1$
System Initialization
1.Generate communication key pairs
Compute $\lambda = \mathrm{lcm}(p-1, q-1)$, $\mu = \left(\mathrm{L}\left(g^{\lambda} \bmod n^2\right)\right)^{-1} \bmod n$;
Keep $CSK = (\lambda, \mu)$ secretly, $CPK = (n, g)$
Publish
1.Generate communication key pairs
Compute $\lambda_e = \mathrm{lcm}(p_e - 1, q_e - 1)$, $\mu_e = \left(\mathrm{L}\left(g_e^{\lambda_e} \bmod n_e^2\right)\right)^{-1} \bmod n_e$;
$CPK_e = (n_e, g_e)$, Keep $CSK_e = (\lambda_e, \mu_e)$ secretly
2.Generate signing key pairs
Register ***PUFs*** ;
Randomly select R_i , compute $PSK = \mathrm{FuzzyExt}(R_i)$;
Randomly select $h \in \mathbb{Z}_{n^2}^{\times}$, select η ;
Compute $ppk = h^{PSK} \bmod n^2$;
$PPK = (ppk, h, \eta)$
Construct & Publish
2.Generate signing key pairs
Register ***PUFs*** ;
Randomly select R_e , compute $PSK_e = \mathrm{FuzzyExt}(R_e)$;
Randomly select $h_e \in \mathbb{Z}_{n_e^2}^{\times}$, select η_e;
Compute $ppk_e = h_e^{PSK_e} \bmod n_e^2$;
$PPK_e = (ppk_e, h_e, \eta_e)$
Key Genernation

Fig. 2. The detailed process of the initialization and key generation phases.

Each ICN and EPN independently generates large primes p and q with the same κ. Then, they compute $n = pq$ and $g = n + 1$, respectively, where g generates a cyclic subgroup of order n modulo n^2.

Additionally, during offline or low-load periods, the ICN performs seed preselection $\theta \leftarrow \{0,1\}^l$ based on κ and constructs a seed preselection pool Θ. This preselected seed pool moves randomness generation away from the time-critical signcryption path on resource-constrained ICNs, enabling $O(1)$ time seed selection, which stabilizes latency and reduces variability under real-time constraints. For each signcryption session, a fresh θ is sampled from Θ to provide per-message freshness without introducing additional communication rounds.

(2) *Key Genernation Phase*

- $ComKeyGen(n, g) \to (CSK, CPK)$

EPN and ICN compute $\lambda = \mathrm{lcm}(p-1, q-1)$ and $\mu = \left(\mathrm{L}(g^{\lambda} \bmod n^2)\right)^{-1} \bmod n$, where $\mathrm{L}(u) = \frac{u-1}{n}$. Each node subsequently stores (λ, μ) as its communication private key CSK and publishes (n, g) as the communication public key.

- $SignKeyGen(GP, R_i) \to (PSK, PPK)$

ICN and EPN register PUFs at the factory, where each device has $C_i \in \{C_1, C_2, ..., C_n\}$ and $R_i \in \{R_1, R_2, ..., R_n\}$ with $R_i = \mathrm{PUF}(C_i)$. Each node randomly selects R_i and computes $PSK = \mathrm{FuzzyExt}(R_i)$ as the signing private key.

Each ICN and EPN then independently chooses $h \in \mathbb{Z}_{n^2}^{\times}$ and selects the large prime order η according to the security parameter κ so that $\eta \,|\, \mathrm{Carm}(n^2)$, where h needs to satisfy $h^n \not\equiv 1 \bmod n^2$ and $h^{\frac{\mathrm{Carm}(n^2)}{\eta}} \not\equiv 1$. They compute $ppk = h^{PSK} \bmod n^2$ and publish $PPK = (ppk, h, \eta)$ as their signing public key.

Owing to their PUF properties, the ICN and EPN can regenerate keys when needed without long-term private key storage.

(3) *Signcryption Phase*

We select the industrial data m collected by ICN_s and the inferred result sent to ICN_r, which is linearly computed by EPN, as an example to illustrate the details of the scheme in the signcryption phase.

- $SCOnline(GP, \theta, CPK_r, PSK_s, m) \to SC$

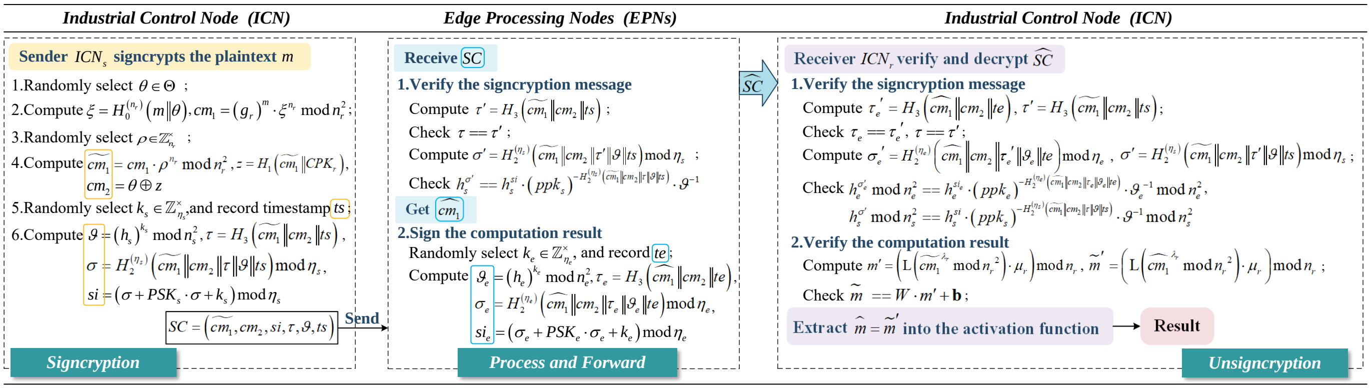

Industrial Control Node (ICN)
Edge Processing Nodes (EPNs)
Industrial Control Node (ICN)
Sender ICN_s signcrypts the plaintext m
1.Randomly select $\theta \in \Theta$;
2.Compute $\xi = H_0^{(n_r)}(m \| \theta), cm_1 = (g_r)^m \cdot \xi^{n_r} \bmod n_r^2$;
3.Randomly select $\rho \in \mathbb{Z}_{n_r}^{\times}$;
4.Compute $\widetilde{cm_1} = cm_1 \cdot \rho^{n_r} \bmod n_r^2, z = H_1(\widetilde{cm_1} \| CPK_r)$, $cm_2 = \theta \oplus z$
5.Randomly select $k_s \in \mathbb{Z}_{\eta_s}^{\times}$,and record timestamp ts;
6.Compute $\vartheta = (h_s)^{k_s} \bmod n_s^2, \tau = H_3(\widetilde{cm_1} \| cm_2 \| ts)$, $\sigma = H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau \| \vartheta \| ts) \bmod \eta_s$, $si = (\sigma + PSK_s \cdot \sigma + k_s) \bmod \eta_s$
$SC = (\widetilde{cm_1}, cm_2, si, \tau, \vartheta, ts)$
Send
Signcryption
Receive SC
1.Verify the signcryption message
Compute $\tau' = H_3(\widetilde{cm_1} \| cm_2 \| ts)$;
Check $\tau == \tau'$;
Compute $\sigma' = H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau' \| \vartheta \| ts) \bmod \eta_s$;
Check $h_s^{\sigma'} == h_s^{si} \cdot (ppk_s)^{-H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau \| \vartheta \| ts)} \cdot \vartheta^{-1}$
Get $\widehat{cm_1}$
2.Sign the computation result
Randomly select $k_e \in \mathbb{Z}_{\eta_e}^{\times}$, and record te;
Compute $\vartheta_e = (h_e)^{k_e} \bmod n_e^2, \tau_e = H_3(\widehat{cm_1} \| cm_2 \| te)$, $\sigma_e = H_2^{(\eta_e)}(\widehat{cm_1} \| cm_2 \| \tau_e \| \vartheta_e \| te) \bmod \eta_e$, $si_e = (\sigma_e + PSK_e \cdot \sigma_e + k_e) \bmod \eta_e$
Process and Forward
$\widehat{SC}$
Receiver ICN_r verify and decrypt $\widehat{SC}$
1.Verify the signcryption message
Compute $\tau_e' = H_3(\widehat{cm_1} \| cm_2 \| te)$, $\tau' = H_3(\widetilde{cm_1} \| cm_2 \| ts)$;
Check $\tau_e == \tau_e'$, $\tau == \tau'$;
Compute $\sigma_e' = H_2^{(\eta_e)}(\widehat{cm_1} \| cm_2 \| \tau_e' \| \vartheta_e \| te) \bmod \eta_e$, $\sigma' = H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau' \| \vartheta \| ts) \bmod \eta_s$;
Check $h_e^{\sigma_e'} \bmod n_e^2 == h_e^{si_e} \cdot (ppk_e)^{-H_2^{(\eta_e)}(\widehat{cm_1} \| cm_2 \| \tau_e \| \vartheta_e \| te)} \cdot \vartheta_e^{-1} \bmod n_e^2$,
$h_s^{\sigma'} \bmod n_s^2 == h_s^{si} \cdot (ppk_s)^{-H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau \| \vartheta \| ts)} \cdot \vartheta^{-1} \bmod n_s^2$
2.Verify the computation result
Compute $m' = \left(\mathrm{L}\left(\widetilde{cm_1}^{\lambda_r} \bmod n_r^2\right) \cdot \mu_r\right) \bmod n_r$, $\widetilde{m}' = \left(\mathrm{L}\left(\widehat{cm_1}^{\lambda_r} \bmod n_r^2\right) \cdot \mu_r\right) \bmod n_r$;
Check $\widetilde{m} == W \cdot m' + \mathbf{b}$;
Extract $\widehat{m} = \widetilde{m}'$ into the activation function
Result
Unsigncryption

Fig. 3. The detailed process of the signcryption, process and forward, and unsigncryption phases.

1) ICN_s selects $\theta \in \Theta$ and computes $\xi = H_0^{(n_r)}(m \| \theta)$, $cm_1 = (g_r)^m \cdot \xi^{n_r} \bmod n_r^2$.
2) ICN_s selects $\rho \in \mathbb{Z}_{n_r}^{\times}$ and computes $\widetilde{cm_1} = cm_1 \cdot \rho^{n_r} \bmod n_r^2$, $z = H_1(\widetilde{cm_1} \| CPK_r)$, $cm_2 = \theta \oplus z$.
3) ICN_s takes $\widetilde{cm_1}$ for subsequent linear operations.
4) ICN_s randomly selects $k_s \in \mathbb{Z}_{\eta_s}^{\times}$ and records the timestamp ts.
5) ICN_s computes the signcryption parameters with Eq. (5), constructs the signcryption message $SC = (\widetilde{cm_1}, cm_2, si, \tau, \vartheta, ts)$, and sends it to EPN for linear computation in private inference.

$$\begin{cases} \vartheta = (h_s)^{k_s} \bmod n_s^2 \\ \tau = H_3(\widetilde{cm_1} \| cm_2 \| ts) \\ \sigma = H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau \| \vartheta \| ts) \bmod \eta_s \\ si = (\sigma + PSK_s \cdot \sigma + k_s) \bmod \eta_s \end{cases} \tag{5}$$

(4) *Process and Forward Phase*

In LAHENet, EPN operates without the decryption private key and performs homomorphic computations on the ciphertext. It verifies the signature of the incoming data before the computation to ensure integrity and resigns the result after the linear computation.

- $LCVerifyOnline(GP, SC, PSK_e, PPK_s) \rightarrow \widehat{SC}$ or $\perp$
  1) After EPN receives the signcryption message SC, it computes $\tau' = H_3(\widetilde{cm_1} \| cm_2 \| ts)$ and verifies whether τ and τ' are equal.
  2) If they are verified as equal, EPN continues to compute $\sigma' = H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau' \| \vartheta \| ts) \bmod \eta_s$. Otherwise, the algorithm outputs $\perp$.
  3) EPN verifies whether $h_s^{\sigma'}$ and $h_s^{si} \cdot (ppk_s)^{-H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau \| \vartheta \| ts)} \cdot \vartheta^{-1}$ are equal in modulus n_s^2.
  4) If they are verified as equal, EPN extracts $\widetilde{cm_1}$ as the ciphertext input for lightweight linear computation in Section IV-D. Otherwise, the algorithm outputs $\perp$.
  5) After linear computation in the cipher domain has been completed, EPN performs a second signature on the result $\widehat{cm_1}$ using its PSK_e.
  6) EPN randomly selects $k_e \in \mathbb{Z}_{\eta_e}^{\times}$ and records the timestamp te.
  7) EPN computes the signcryption parameters with Eq. (6), constructs the signcryption message $\widehat{SC} = (\widehat{cm_1}, SC, si_e, \tau_e, \vartheta_e, te)$, and sends it to ICN_r for nonlinear activation.

$$\begin{cases} \vartheta_e = (h_e)^{k_e} \bmod n_e^2 \\ \tau_e = H_3(\widehat{cm_1} \| cm_2 \| te) \\ \sigma_e = H_2^{(\eta_e)}(\widehat{cm_1} \| cm_2 \| \tau_e \| \vartheta_e \| te) \bmod \eta_e \\ si_e = (\sigma_e + PSK_e \cdot \sigma_e + k_e) \bmod \eta_e \end{cases} \tag{6}$$

(5) *UnSigncryption Phase*

- $USCOnline\left(GP, \widehat{SC}, CSK_r, CPK_r, CPK_e, PPK_s, PPK_e\right) \rightarrow \widehat{m}$ or $\perp$
  1) After ICN_r receives the processed message $\widehat{SC}$, it computes $\tau_e' = H_3(\widehat{cm_1} \| cm_2 \| te)$, $\tau' = H_3(\widetilde{cm_1} \| cm_2 \| ts)$ and first verifies whether τ_e and τ_e', τ and τ' are equal.
  2) If they are verified as equal, ICN_r computes $\sigma_e' = H_2^{(\eta_e)}(\widehat{cm_1} \| cm_2 \| \tau_e' \| \vartheta_e \| te) \bmod \eta_e$, $\sigma' = H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau' \| \vartheta \| ts) \bmod \eta_s$. Otherwise, the algorithm outputs $\perp$.
  3) ICN_r verifies whether $h_e^{\sigma_e'}$ and $h_e^{si_e} \cdot (ppk_e)^{-H_2^{(\eta_e)}(\widehat{cm_1} \| cm_2 \| \tau_e \| \vartheta_e \| te)} \cdot \vartheta_e^{-1}$ in modulus n_e^2, $h_s^{\sigma'}$ and $h_s^{si} \cdot (ppk_s)^{-H_2^{(\eta_s)}(\widetilde{cm_1} \| cm_2 \| \tau \| \vartheta \| ts)} \cdot \vartheta^{-1}$ in modulus n_s^2 are equal.
  4) If both are verified as equal, ICN_r performs the final inference computation and verifies the results. Otherwise, the algorithm outputs $\perp$.
  5) ICN_r computes with Eq. (7) and verifies whether $\widetilde{m}'$ and $W \cdot m' + \mathbf{b}$ are equal to check the computational fraud in EPN.

$$\begin{cases} m' = \left(\mathrm{L}\left(\widetilde{cm_1}^{\lambda_r} \bmod {n_r}^2\right) \cdot \mu_r\right) \bmod n_r \\ \widetilde{m}' = \left(\mathrm{L}\left(\widehat{cm_1}^{\lambda_r} \bmod {n_r}^2\right) \cdot \mu_r\right) \bmod n_r \end{cases} \tag{7}$$

  6) If they are verified as equal, ICN_r computes $z' = H_1(\widetilde{cm_1} \| CPK_r)$ and $\theta' = cm_2 \oplus z'$ according to the log auditing requirements. Otherwise, it indicates that there is computational fraud in EPN, and the algorithm outputs $\perp$ immediately.
  7) Finally, ICN_r extracts $\widehat{m} = \widetilde{m}'$ into the nonlinear activation function to obtain the inference result.

D. Lightweight Linear Computation Approach

Given a multilayer neural network, $W^{(\ell)} \in \mathbb{Z}^{h_\ell \times d_{\ell-1}}$, $\mathbf{b}^{(\ell)} \in \mathbb{Z}^{h_\ell}$ in the ℓ-th layer, where $\ell = 1, ..., L$, d_0 is the input dimension, and $d_{\ell-1}$ and h_ℓ represent the input and output widths of the ℓ-th layer, respectively.

LAHENet performs a single-pass aggregation strategy that completes all linear mappings in the ciphertext domain at the IIoT edge. ICN_s encrypts each component of the plaintext feature vector $\mathbf{m}$ and signs the ciphertext stream in Eq. (8).

$$\mathbf{c} = \left\{ \widetilde{cm_1^1}, \widetilde{cm_1^2}, ..., \widetilde{cm_1^{d_0}} \right\} \in \mathbb{Z}_{n_r^2}^{\times} \tag{8}$$

This stream serves as input to EPN, which performs all linear mappings on the ciphertext domain in a single pass and transmits only the final ciphertext to ICN_r to reduce the interaction overhead in conventional private inference.

Multiple linear layers can be merged into a single large matrix operation via adaptive layer folding, or homomorphic accumulation can be performed sequentially by layer. This ensures that ICN_r performs only a single decryption before the nonlinear activation function is applied.

In the offline phase, LAHENet constructs an edge device information triplet $(ERAM, EMODS, ICSLOT)$ when it compiles, where $ERAM$ represents the available memory size of EPN, $EMODS$ represents the throughput per second of EPN continuously performing matrix multiplication, and $ICSLOT$ is the single-cycle real-time budget of each ICN that belongs to EPN. EPN constructs a single-layer folding strategy evaluation model with Eq. (9) and checks whether $\sum\limits_{\ell=1}^{L} \mathcal{B}_\ell \leq \frac{1}{2} ERAM$, $\max_\ell \mathcal{R}_\ell \leq 0.7 ICSLOT$ during the operation cycle.

$$\begin{cases} \mathcal{B}_\ell = h_\ell \left(2 \left| n_r \right| + q/8 \right) \\ \mathcal{R}_\ell = h_\ell d_{\ell-1} / EMODS \end{cases} \tag{9}$$

If both conditions hold, the L set $(W_\ell, \mathbf{b}_\ell)$ is folded into a single layer using Eq. (10), where $\overline{W} \in \mathbb{Z}^{h_L \times d_0}$, $\overline{\mathbf{b}} \in \mathbb{Z}^{h_L}$. This folded layer is then used in online inference. Otherwise, the weights and accumulators are processed in a layer-by-layer serial mode during online inference.

$$\begin{cases} \overline{W} = W_L \cdots W_1 \\ \overline{\mathbf{b}} = W_L \cdots W_2 \mathbf{b}_1 + W_L \cdots W_3 \mathbf{b}_2 + ... + \mathbf{b}_L \end{cases} \tag{10}$$

In the online phase, EPN initializes a zeroing accumulator $A_j^{(\ell,0)}$ for each output unit j in layer ℓ, as defined in Eq. (11). Then, it iterates over the input ciphertexts $\left\{ \widetilde{cm_1^i} \right\}_{i=1}^{d_{\ell-1}}$ of that layer.

$$A_j^{(\ell,0)} = g_r^0 \bmod n_r^2, j = 1, ..., h_\ell \tag{11}$$

The weight matrix elements satisfy $w_{ji}^{(\ell)} \in \left[-2^{q-1}, 2^{q-1} - 1 \right]$. If $w_{ji}^{(\ell)} \geq 0$, the exponent $w_{ji}^{(\ell)}$ is used directly. Otherwise, let $\left(\widetilde{cm_1^i} \right)^{w_{ji}^{(\ell)}} = \left(\widetilde{cm_1^i} \right)^{n_r + w_{ji}^{(\ell)}} \bmod n_r^2$ and perform accumulation with Eq. (12), Eq. (13), where $\oplus$ represents homomorphic addition in the cipher domain and $\odot$ represents scalar multiplication in the cipher domain.

$$A_j^{(\ell,i)} = A_j^{(\ell,i-1)} \oplus \left(\widetilde{cm_1^i} \odot e_{ji}^{(\ell)} \right) \tag{12}$$

$$e_{ji}^{(\ell)} = \begin{cases} w_{ji}^{(\ell)}, & w_{ji}^{(\ell)} \geq 0 \\ n_r + w_{ji}^{(\ell)}, otherwise \end{cases} \tag{13}$$

After traversing all the inputs in layer ℓ, EPN obtains Eq. (14) and computes $A_{\ell,j}^{*} = A_j^{(\ell,d_{\ell-1})} \oplus (g_r \odot b_{\ell,j})$, where $Cp_i^{(\ell-1)}$ represents the i-th ciphertext of the input in this layer. For the first layer ($\ell = 1$), $Cp_i^{(0)} = \widetilde{cm_1^i}$ and $Cp_i^{(\ell-1)} = A_{\ell-1,j}^{*}$ for the deeper layer ($\ell > 1$).

$$A_j^{(\ell,d_{\ell-1})} = \prod_{i=1}^{d_{\ell-1}} \left(Cp_i^{(\ell-1)} \right)^{w_{ji}^{(\ell)}} \bmod n_r^2 \tag{14}$$

Additionally, LAHENet incorporates a ciphertext exponentiation accelerator (CEA) for the online phase to further optimize inference performance in IIoT edge environments. As shown in Algorithm 1, the CEA is specifically designed to handle the computationally intensive exponentiation operations in the ciphertext domain. By precomputing and efficiently indexing exponentiation results, it substantially reduces the online computational cost and latency, enabling real-time inference while preserving homomorphic security guarantees.

Algorithm 1 : Ciphertext Exponentiation Accelerator (CEA)

Input: $\{\widetilde{cm_1^i}\}_{i=1}^{d_{\ell-1}}$, $w_{ji}^{(\ell)} \in [-2^{q-1}, 2^{q-1} - 1]$, and window size $\mathcal{K}$.
Output: $(\widetilde{cm_1^i})^{e_{ji}^{(\ell)}}$.

```
    Offline Phase
for $\widetilde{cm_1^i}$ in $\{\widetilde{cm_1^i}\}_{i=1}^{d_{\ell-1}}$ do
    $num\_windows \leftarrow \lceil q/\mathcal{K} \rceil$
    for $j \leftarrow 0$ to $num\_windows - 1$ do
        $B_j \leftarrow \left(\widetilde{cm_1^i}\right)^{2^{j-\mathcal{K}}} \bmod n_r^2$
        for $\mathcal{P} \leftarrow 0$ to $2^{\mathcal{K}} - 1$ do
            $\mathcal{Q}_{i,j}\left[\mathcal{P}\right] \leftarrow B_j^{\mathcal{P}} \bmod n_r^2$
        end for
    end for
end for
    Online Phase
if $w_{ji}^{(\ell)} < 0$ then
    $e_{ji}^{(\ell)} \leftarrow n_r + w_{ji}^{(\ell)}$
else
    $e_{ji}^{(\ell)} \leftarrow w_{ji}^{(\ell)}$
end if
$Result \leftarrow 1 \bmod n_r^2$
for $j \leftarrow 0$ to $num\_windows - 1$ do
    $w_j \leftarrow \left(e_{ji}^{(\ell)} \gg (j \cdot \mathcal{K})\right) \ \& \ \left(2^{\mathcal{K}} - 1\right)$
    Search $Term_j \leftarrow \mathcal{Q}_{i,j}\left[w_j\right]$
    $Result \leftarrow Result \cdot Term_j \bmod n_r^2$
end for
return $Result$
```

Correctness of LAHENet: We assume that the data sender is ICN_s, the linear computation processor is EPN_e, and the data receiver is ICN_r. Under the LAHENet framework, if all nodes execute the protocol honestly, then for any plaintext feature vector $\mathbf{m} = (m_1, ..., m_{d_0})^{\mathrm{T}} \in \mathbb{Z}_{n_r}^{d_0}$ and a publicly available linear model $\{W_\ell, \mathbf{b}_\ell\}_{\ell=1}^{L}$, ICN_r obtains the result

$\widehat{\mathbf{m}}$ after executing the $USCOnline$. This result is exactly equal to the expected output of applying the specified linear computation to the original plaintext $\mathbf{m}$, as in Eq. (15). $\overline{W}$ and $\overline{\mathbf{b}}$ are obtained either by the folded composition in Eq. (10) or by the layer-wise accumulation in Eq. (13). Both constructions yield the same linear form under identical numeric precision and quantization, because folding changes only the computational path and does not alter the numerical output or the model accuracy.

$$\widehat{\mathbf{m}} = \begin{cases} \mathbf{m}, & L = 0 \\ \overline{W}\mathbf{m} + \overline{\mathbf{b}}, L \geq 1 \end{cases} \tag{15}$$

ICN_s signcrypts m using the random seed θ and a random mask $\rho \in \mathbb{Z}_{n_r}^{\times}$ to construct $\widetilde{cm_1} = (g_r)^m \cdot \xi^{n_r} \cdot \rho^{n_r} \bmod n_r^2$ in $SCOnline$, where $\xi = H_0^{(n_r)}(m \| \theta) \in \mathbb{Z}_{n_r}^{\times}$. Given that both ξ^{n_r} and ρ^{n_r} are n_r-subresidues, it follows Eq. (16).

$$\begin{aligned} (\widetilde{cm_1})^{\lambda_r} &= ((g_r)^m)^{\lambda_r} \cdot (\xi^{n_r})^{\lambda_r} \cdot (\rho^{n_r})^{\lambda_r} \\ &\equiv (g_r)^{m\lambda_r} \equiv (1+n_r)^{m\lambda_r} \bmod n_r^2 \end{aligned} \tag{16}$$

For $L = 0$, we have $(1+n_r)^{m\lambda_r} \equiv 1 + mn_r\lambda_r \bmod n_r^2$ by Eq. (16). ICN_r performs the following computation, which demonstrates that the ciphertext without linear computation can be correctly decrypted to the original plaintext under honest execution of the protocol.

$$\begin{aligned} m' &= \mathrm{L}\left((\widetilde{cm_1})^{\lambda_1} \bmod n_r^2\right) - \mu_r \bmod n_r \\ &= \frac{(1 + mn_r\lambda_r) - 1}{n_r} - \mu_r \bmod n_r \\ &= m\lambda_r\mu_r \bmod n_r = m \bmod n_r \end{aligned}$$

For $L \geq 1$, let $\widetilde{\mathbf{c}}^{(0)} = \left(\widetilde{cm_1^1}, ..., \widetilde{cm_1^{d_0}}\right)$, $\widetilde{cm_1^i} = (g_r)^{m_i} \cdot \xi_i^{n_r} \cdot \rho_i^{n_r} \bmod n_r^2$. EPN_e performs a linear computation in the ciphertext domain. For the output unit j, we define $\left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j = (W_L \cdots W_1\mathbf{m})_j + \sum_{\ell=1}^{L} W_L \cdots W_{\ell+1}\mathbf{b}_\ell$. The result produced by these series of linear operations is $\widehat{cm_1} = (g_r)^{\left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j} \left(\prod_{i=1}^{d_0} \xi_i^{\alpha_{ji}} \rho_i^{\alpha_{ji}}\right)^{n_r} \bmod n_r^2$, where α_{ji} is the integer coefficient of m_i in this expansion.

Subsequently, we establish $\Re = \prod_{i=1}^{d_0} \xi_i^{\alpha_{ji}} \rho_i^{\alpha_{ji}}$ to simplify the expression derived. ICN_r performs the following computation on the result, which demonstrates that ICN_r can still correctly decrypt the ciphertext even after the linear operations on the ciphertext domain.

$$\begin{aligned} m_j' &= \mathrm{L}\left((\widetilde{cm_1})^{\lambda_r} \bmod n_r^2\right) \cdot \mu_r \bmod n_r \\ &= \mathrm{L}\left((g_r)^{\lambda_r\left(\left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j\right)} \Re^{\lambda_r n_r}\right) \cdot \mu_r \bmod n_r \\ &= \mathrm{L}\left((1+n_r)^{\lambda_r\left(\left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j\right)}\right) \cdot \mu_r \bmod n_r \\ &= \frac{(1+n_r)^{\lambda_r\left(\left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j\right)} - 1}{n_r} \cdot \mu_r \bmod n_r \\ &= \frac{\left(1 + n_r\lambda_r\left(\left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j\right)\right) - 1}{n_r} \cdot \mu_r \bmod n_r \\ &= \left(\left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j\right)\lambda_r\mu_r \bmod n_r = \left(\overline{W}\mathbf{m}\right)_j + \bar{b}_j \bmod n_r \end{aligned}$$

Therefore, we conclude that the LAHENet framework achieves correctness in its core computational functionality under honest execution of the protocol.

V. Security Analysis

In this section, we analyse the security of the LAHENet framework through a unified security experiment, which are given in *Appendix A*. Detailed proofs concerning the confidentiality, unforgeability, forward security, and verifiable computational correctness of the LAHENet framework are demonstrated in *Appendix B*. The adversary $\mathcal{A}$ combines the capabilities of $\mathcal{A}_{net}$ and $\mathcal{A}_{node}$ defined in the threat model (Section III-B). Finally, we discuss the practical security considerations and limitations of the framework in Section V-E.

A. Confidentiality

Theorem 1 (IND-CCA2-LAHENet): For private inference in the IIoT, the signcryption protocol of the LAHENet framework satisfies confidentiality under adaptive chosen ciphertext attacks when no $\mathcal{A}$ can distinguish between two signcryptions of equal-length messages with a nonnegligible advantage.

Proof: The advantage of an adversary $\mathcal{A}$ in breaking $IND-CCA2-LAHENet$ is given by Eq. (17), where q_{usc}, q_{sc}, q_0, and q_1 are the numbers of queries to the oracles O_{USC}, O_{SC}, H_0, and H_1. The ε_{sig} is the maximum advantage in successful forging node signatures, and $Adv_{\mathcal{A}}^{DCR}(\kappa)$ is the maximum advantage in solving the DCR problem. The detail proof is given in *Appendix B*.

$$\begin{aligned} &Adv_{\Omega,\mathcal{A}}^{IND-CCA2-LAHENet} \leq \\ &(q_{usc} + q_{sc}) \cdot \varepsilon_{sig} + \frac{(q_0 + q_1)^2}{2^{|n_r|}} + Adv_{\mathcal{A}}^{DCR}(\kappa) \end{aligned} \tag{17}$$

B. Unforgeability

Theorem 2 (EUF-CMA-LAHENet): In the private inference process of the IIoT, the signcryption protocol of the LAHENet framework satisfies existential unforgeability under adaptive chosen message attacks when no $\mathcal{A}$ can forge valid signcryption on any new message with a nonnegligible advantage unless it can solve the DLP in polynomial time.

Proof: The advantage of an adversary $\mathcal{A}$ in breaking $EUF-CMA-LAHENet$ is given by Eq. (18), where q_{sc}, q_2, and q_3 are the numbers of queries to the oracles O_{SC}, H_2, and H_3, respectively. η_s and l_{tag} denote the size and length of the output space, respectively, and $Adv_{\mathcal{C}}^{DLP}(\kappa)$ is the maximum advantage in solving the DLP. The detail proof is given in *Appendix B*.

$$\begin{aligned} & Adv_{\Omega,\mathcal{A}}^{EUF-CMA-LAHENet} \le \\ & \frac{q_{sc}\cdot q_2}{|\eta_s|} + \frac{(q_{sc}+q_2)^2}{2|\eta_s|} + \frac{q_3^2}{2^{l_{tag}}} + Adv_{\mathcal{C}}^{DLP}(\kappa) \end{aligned} \tag{18}$$

C. Forward Security

Theorem 3 (FwSec-LAHENet): For private inference in the IIoT, the signcryption protocol of the LAHENet framework satisfies forward security. This means that even if $\mathcal{A}$ corrupts the sender at the future time T and obtains its private key, it still cannot distinguish any signcryption message generated at $ts < T$ nor forge a valid signcryption corresponding to a timestamp $ts < T$ with a nonnegligible advantage.

Proof: The advantage of an adversary $\mathcal{A}$ in breaking $FwSec-LAHENet$ is given by Eq. (19), where q_{usc}, q_{sc}, q_0, q_1, q_2, and q_3 are the numbers of queries to the oracles O_{USC}, O_{SC}, H_0, H_1, H_2, and H_3, respectively. ε_{sig} is the maximum advantage in successful forging node signatures, and η_s and l_{tag} denote the size and length of the output space, respectively. $Adv_{\mathcal{A}}^{DCR}(\kappa)$ and $Adv_{\mathcal{C}}^{DLP}(\kappa)$ are the maximum advantages in solving the DCR and DLP, respectively. The detail proof is given in *Appendix B*.

$$\begin{aligned} & Adv_{\Omega,\mathcal{A}}^{FwSec-LAHENet} \le \\ & (q_{usc}+q_{sc})\cdot\varepsilon_{sig} + \frac{q_{sc}\cdot q_2}{|\eta_s|} + \frac{q_{sc}\cdot q_3}{2^{l_{tag}}} + \frac{(q_0+q_1)^2}{2^{|n_r|}} \\ & + Adv_{\mathcal{C}}^{DLP}(\kappa) + Adv_{\mathcal{A}}^{DCR}(\kappa) \end{aligned} \tag{19}$$

D. Verifiable Computational Correctness

Theorem 4 (VerCorrect-LAHENet): For private inference in the IIoT, the inference of the LAHENet framework satisfies verifiable computational correctness when no $\mathcal{A}$ can induce the receiver, with a nonnegligible advantage, to accept a linear inference result that is inconsistent with the public model $(W, \mathbf{b})$.

Proof: The advantage of an adversary $\mathcal{A}$ succeeding in making the receiver accept an incorrect result of a linear computation is given by Eq. (20), where $Adv_{\Omega,\mathcal{A}}^{EUF-CMA-LAHENet}$ has been proven in *Theorem 2*. The detail proof is given in *Appendix B*.

$$\begin{aligned} & Adv_{\Omega,\mathcal{A}}^{VerCorrect-LAHENet} \\ & \le Adv_{\Omega,\mathcal{A}}^{EUF-CMA-LAHENet} + \mathrm{negl}(\kappa) \end{aligned} \tag{20}$$

E. Security Discussion

Although LAHENet has proven secure under the random oracle model for confidentiality, unforgeability, forward security, and verifiable computation, we still need to consider some limitations when deploying it in real IIoT.

First, the reliance on PUFs leaves the framework potentially vulnerable to adversaries with advanced physical intrusion or side-channel analysis capabilities. Second, LAHENet targets IIoT edge deployments in which the inference model is typically transparent within a controlled domain. Under this public-model assumption, a single pre-activation is revealed only to the legitimate receiver for local consistency verification, while the edge processing node remains confined to the ciphertext domain. Consequently, the data-privacy attack surface is not enlarged. While prior work, such as nGraph-HE2 [24], correctly notes that client-aided activations pose a risk of model extraction when model secrecy is a goal, this risk does not apply under our threat model, which focuses on data confidentiality and low-latency verifiable correctness for industrial control.

In deployments where model confidentiality is also required, heavyweight primitives like GC or TEEs may be adopted to provide protected activation. Such mechanisms typically introduce higher communication overhead and latency, thereby affecting the efficiency goals of this lightweight framework. Furthermore, we have not focused on Machine learning (ML) specific threats in this paper, such as adversarial examples, which may require measures of differential privacy or robustness training in deployments.

These limitations do not weaken the cryptographic security of the proposed framework but rather provide direction for our future work. In future work, we will focus on PUF-hardening mechanisms that are resilient to invasive attacks and robust ML defences, such as certified adversarial robustness. Meanwhile, extending LAHENet to private-model deployments that require model confidentiality is orthogonal to our present focus and is left for future work.

VI. Performance Experiments

A. Experimental Setup

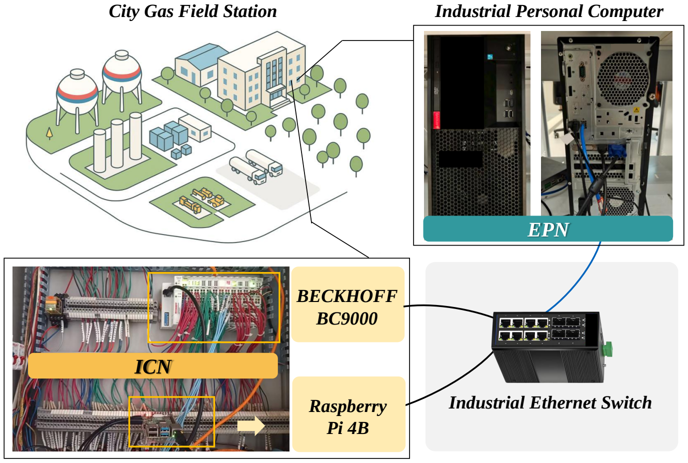
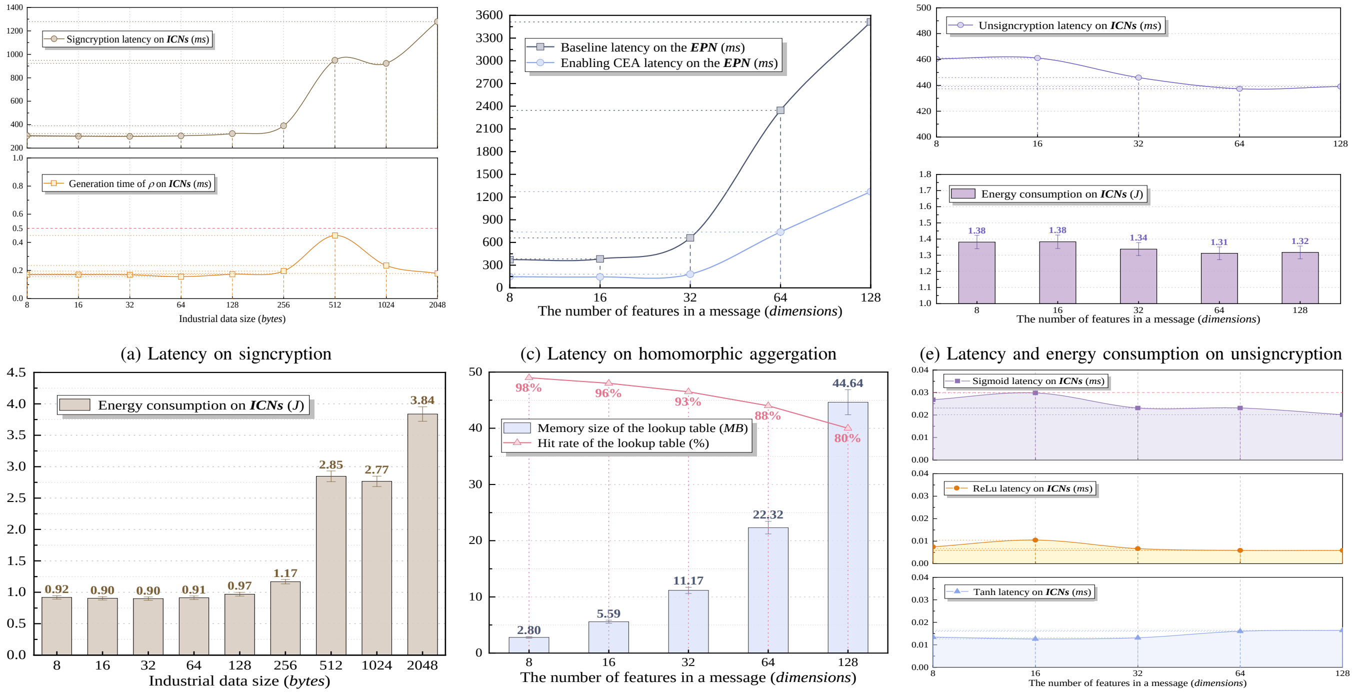


Fig. 4. Experimental environments and devices within station.

To evaluate the performance of the LAHENet framework comprehensively in a real IIoT environment and confirm its suitability for strict computational resource, power consumption, and device heterogeneity, we deploy a prototype in the



(a) Latency on signcryption

(c) Latency on homomorphic aggergation

(e) Latency and energy consumption on unsigncryption

(b) Energy consumption on signcryption

(d) Memory size and hit rate of the lookup table

(f) Latency on different activate functions

Fig. 5. Performance of LAHENet under different phases.

production network of a city gas field station. As depicted in Fig. 4, we combine the BECKHOFF BC9000 of this station with the Raspberry Pi 4B as ICNs and select an Industrial personal computer (IPC) with Intel i7-8700 CPUs, 16 GB RAM, and Windows 10 as the EPN. During the experiment, ICNs periodically poll and read data from device registers via Modbus TCP to perform real-time data collection and preprocessing at the site, and the EPN executes core privacy inference tasks of the LAHENet framework. Its hardware configuration represents the practical computational capabilities currently achievable at the IIoT edge.

Implementation and Datasets. The LAHENet framework is implemented in Python 3.9 and PyTorch 2.3.1. We perform all the model training and inference tasks on the CPU due to the lack of GPU support on IIoT devices. For an objective evaluation, we conduct experiments over a private and a public dataset as follows. Each dataset is randomly divided into an 80% training set and a 20% testing set, and the initial learning rate for model training is set to 0.001, with a mini-batch size of 64. We employ a Multilayer perceptron (MLP) as the task model. In Section VI-B, we set the network depth to four (counting linear layers) to obtain stable component-wise latency and overhead measurements. In Section VI-C, we use a network depth of eight to reflect a representative mid-depth edge deployment for system-level evaluation.

- *CG Dataset*: A private dataset collected from the SCADA system at this station. It contains 50,000 valid samples over three consecutive months, covering parameters such as current, voltage, power, and temperature. We use this dataset to validate the effectiveness of LAHENet with $\delta = 0.5\%$ in prediction tasks, and to ground our evaluation in real industrial deployment scenarios.
- *SWaT Dataset* [49]: A publicly available industrial control system dataset from a real-world water treatment testing platform. It contains more than 400,000 normal and 54,621 attack samples. We use this dataset to evaluate the anomaly detection performance of LAHENet with $\delta = 1.0\%$ across industry scenarios, and validate the applicability of our framework for different industrial tasks.

B. Component-Level Performance Evaluation

1) Signcryption Overhead on Resource-Constrained Nodes

We first evaluated the signcryption latency and energy consumption of LAHENet on resource-constrained ICNs, testing message sizes from 8 to 2048 bytes.

As shown in Fig. 5a, for typical industrial payloads (8 to 128 bytes), the average latency remains stable at approximately 300 ms. This stability stems from the fixed computational cost of modular exponentiation, whereas the message content contributes only a constant number of multiplication operations, exerting a negligible influence on the overall delay. When the message length exceeds 512 bytes, the latency increases nearly linearly due to block processing. However, it remains under 1300 ms even for a 2048-byte payload. It is applicable for many critical IIoT monitoring tasks, which typically operate on cycles of one second or longer. Notably, the generation of the random mask ρ required less than 0.45 ms and under 0.06% of the total latency, which confirms that the security mechanism of LAHENet does not constitute a bottleneck for resource-constrained nodes.

Additionally, Fig. 5b shows the energy consumption for single signcryption ranged from 0.9-3.8 J. Considering that industrial data sampling cycles generally last more than one second, this energy usage is well within the acceptable power budget for edge devices powered by local or uninterruptible supplies.

2) Homomorphic Aggregation Performance at the Edge

The efficiency of homomorphic aggregation on the EPN is a decisive factor in LAHENet's throughput, particularly as feature dimensionality increases. We therefore measured aggregation latency for dimensions ranging from 8-128. As shown in Fig. 5c, the baseline implementation exhibits sharp, superlinear growth in latency, increasing from 371.4 ms at 8 dimensions to a prohibitive 3513 ms at 128 dimensions. This underscores the critical need for targeted optimization in IIoT.

The CEA in LAHNet eliminates the aggregation bottleneck by constructing an exponential lookup table for each input ciphertext during the offline phase and replacing expensive modular exponentiations with $O(1)$ table lookups online. The results demonstrate that CEA significantly reduces latency across all evaluated dimensions, achieving a 3.66x increase in speed at 32 dimensions. Although the acceleration remains notable, the acceleration gradually decreases to 3.18x at 64 dimensions and 2.77x at 128 dimensions.

We analyse the hardware of the EPN in Fig. 5d and believe that this reduction is attributed to a decrease in the cache hit rate. When the feature dimension increases to 64 and 128, the size of the lookup table exceeds the capacity of the L3 cache, causing the cache hit rate to decrease from 98% to 80%. Cache hits are amortized into total latency, thereby diminishing the overall acceleration. Despite the reduced speed-up at higher dimensions, CEA still trims 2243.1 ms of latency per sample, and its memory footprint at 128 dimensions occupies only 0.35% of the IPC. These well-characterized trade-offs enable efficient and scalable private inference for high-dimensional IIoT workloads on edge hardware.

3) Finalization Stage Overhead at the Receiver

The final stage of the LAHENet framework involves the receiver node performing verification, decryption, and nonlinear activation. To ensure that this stage does not become a scalability bottleneck, we evaluate its performance on ICNs and show the results in Fig. 5e, Fig. 5f.

The unsigncryption latency remains stable between 437.3 ms and 461.1 ms, even if the feature dimension increases from 8 to 128, which is a key design feature of the LAHENet framework. This dimension-independent cost is due to the verification process running on the hash values of fixed-size signatures and aggregated ciphertexts rather than on the variable-length plaintext vectors within them. Moreover, the minor fluctuations observed in the results are attributed to standard experimental jitter on nonreal-time operating systems.

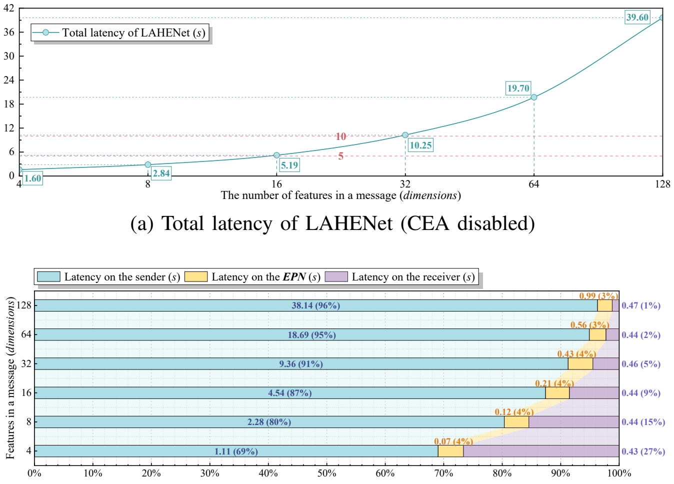
For a 128-dimensional scenario, $Sigmoid$ requires less than 0.03 ms, with other functions such as $ReLU$ and $Tanh$ performing even faster. Consequently, the total time for this stage on the ICN is less than 480 ms and accounts for less than 15% of the typical end-to-end latency, which still satisfies the real-time requirements of the IIoT. Additionally, the energy consumption of a single message remains at the 1.3 J level, posing no burden on industrial power supplies.

These results demonstrate that LAHENet successfully offloads the scalable computational workload to the more powerful EPN, with the receiving end bearing only constant and predictable overhead.

C. System-Level Performance Evaluation

1) Baseline Performance Metrics

We completed a system-level evaluation of LAHENet by establishing benchmark metrics in an ideal single-traffic scenario. Unless otherwise noted, Fig. 6 reports latencies with the CEA disabled to provide a uniform system-level baseline for bottleneck analysis.



(a) Total latency of LAHENet (CEA disabled)

(b) Detailed latency on the sender, EPN, and receiver

Fig. 6. Latency baseline of LAHENet in an ideal scenario.

As shown in Fig. 6a, the total system latency is strongly positively correlated with the input dimension, increasing from 1.6 s for 4 dimensions to 39.61 s for 128 dimensions. A detailed latency decomposition in Fig. 6b clearly identifies the performance bottleneck. The resource-constrained ICN is responsible for 69-96% of the total latency because of the cost of source-side signcryption. In contrast, the EPN accounts for only a small fraction of the total time. These results confirm the successful offloading of computationally intensive tasks to the edge. We also observe that the processing time of the EPN does not grow strictly linearly, which aligns with our component-level analysis and reflects system-level effects such as CPU cache saturation under larger cryptographic contexts. Baseline experiments further show that the load of the receiver ICN is stable and predictable, stabilizing at approximately 0.45 s.

Although the latency of high-dimensional tasks ($d > 32$) exceeds the requirements for hard real-time control, most practical industrial applications rely on a moderate number of features (typically $d < 32$) after feature selection. Scenarios requiring the secure transmission of very high-dimensional data ($d > 64$) are rare and usually limited to forensic analysis or offline diagnostics, where longer processing times are an acceptable trade-off for end-to-end security. Therefore, these baseline results demonstrate that LAHENet is able to provide strong security within the practical operating range of industrial applications.

2) Analysis of Optimization Contribution

We performed an ablation study on LAHENet to evaluate the impact of its core optimizations in Fig. 7. The fully optimized framework serves as the baseline for comparison with variants where each optimization is individually disabled.

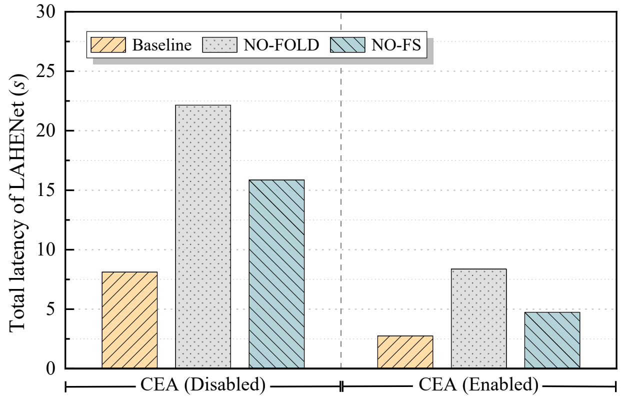


Fig. 7. Comparison the impact of optimization on LAHENet.

When layer folding is disabled (NO-FOLD), the EPN must perform homomorphic aggregation and append signatures layer by layer. This approach multiplies the number of modular exponentiations and communication round trips, increasing the latency to 8.38 s and resulting in a 3x performance degradation. This highlights the efficacy of the layer-folding strategy in minimizing redundant cryptographic operations and intermediate data processing. Similarly, disabling feature selection (NO-FS) requires the ICN to signcrypt the original high-dimensional features, and the overhead of EPN lookup and aggregation also expands. It increases the latency to 4.74 s, a 72% increase over the baseline.

Notably, even if the 2.76 s latency is unsuitable for the 0.5 s requirement of hard real-time control loops, it fully satisfies the 5 s safety-interlock window widely adopted in IIoT. It also aligns with the demands of IIoT monitoring tasks such as predictive maintenance, anomaly detection, and energy optimization, which operate at the second-level granularity. Therefore, LAHENet is practically deployable in safety-critical industrial edge environments.

3) Performance under Stress Conditions

To evaluate the performance and robustness of LAHENet beyond ideal conditions, we evaluated it under two stress scenarios.

- Scalability with Multi-ICN Concurrency

We experiment with the performance of a single EPN while handling up to 64 concurrent ICN clients, each continuously submitting 32-dimensional data vectors. As shown in Fig. 8, the throughput ranges from 0.097 req/s with one client to 2.21 req/s with 32 clients, confirming the capacity of the EPN for parallel request processing. Beyond this point, throughput gains diminish, indicating that the computational resources of the EPN are approaching saturation. When it reaches 2.45 req/s at 64 clients, the sharp increase in latency past 32 clients further confirms this saturation point. Notably, the 99th percentile latency remains low until the system nears this threshold. Additionally, the memory usage remained stable at approximately 57 MB during this experiment, which demonstrates excellent memory stability under high concurrency.

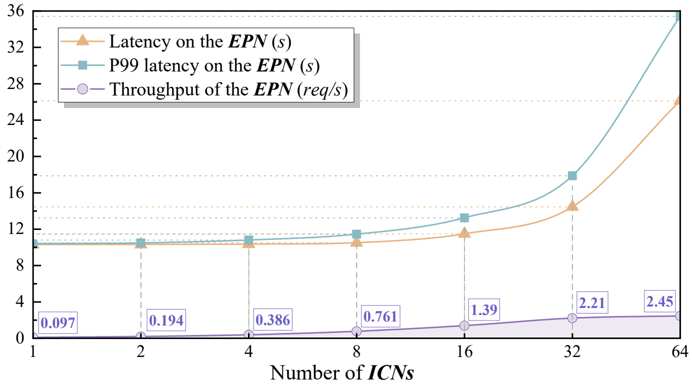


Fig. 8. Latency and throughput under multi ICNs concurrency.

- Robustness over Unreliable Networks

We evaluated system robustness by introducing packet loss rates from 0-10%. As shown in Fig. 9, LAHENet maintains high operational integrity even under adverse network conditions. The transaction success rate remains above 98% even at 10% packet loss. This is because the fast signature-based integrity check of LAHENet allows the receiver to discard corrupted packets before expensive homomorphic operations to avoid wasting computational resources on damaged data. Although the 99th percentile latency is more sensitive, the delay time increases to 37.3 s under a 10% packet loss rate because of retransmission timeouts. However, the high success rate is a more critical metric for operational integrity than the significant increase. These results demonstrate that LAHENet is robust enough for deployment in challenging industrial environments where network stability is not guaranteed.

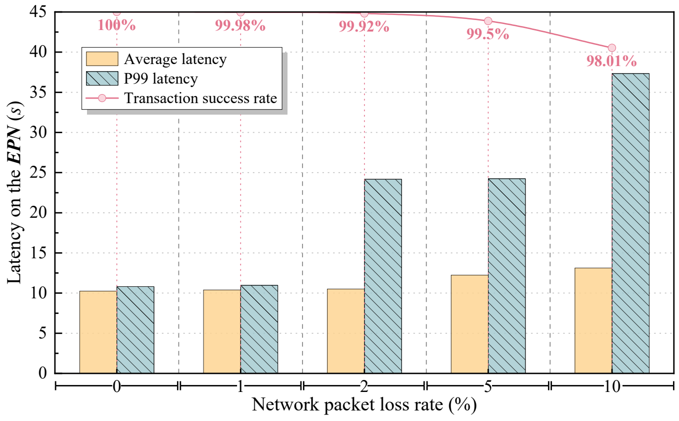


Fig. 9. Latency and success rate under different packet loss rate.

4) Effectiveness in Real-World Industrial Tasks

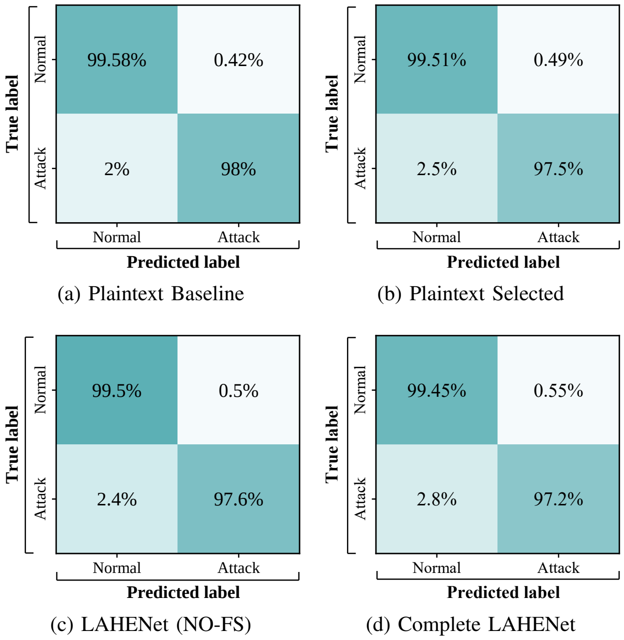
We evaluated the effectiveness of LAHENet on two representative industrial tasks, with the results summarized in Table II. We set up four controls of the plaintext model without feature selection (Plaintext Baseline), the plaintext model with feature selection (Plaintext Selected), the LAHENet framework without feature selection (LAHENet (NO-FS)) and the

TABLE II
COMPARISON WITH PLAINTEXT AND CIPHERTEXT OF LAHENET ON DIFFERENT INDUSTRIAL TASKS

Task	Model	Dimension	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	MAPE (%)	RMSE (kW)	Latency (s)
Anomaly Detection (on *SWaT*)	Plaintext Baseline	51	99.39	97.00	98.00	97.50	-	-	0.45
	Plaintext Selected	25	99.27	96.50	97.50	97.00	-	-	0.34
	LAHENet (NO-FS)	51	99.26	96.40	97.60	97.00	-	-	13.74
	Complete LAHENet	25	99.18	96.10	97.20	96.65	-	-	8.14
Regression Prediction (on *CG*)	Plaintext Baseline	48	-	-	-	-	5.1	6.9	0.43
	Plaintext Selected	27	-	-	-	-	5.3	7.4	0.37
	LAHENet (NO-FS)	48	-	-	-	-	5.8	7.7	12.94
	Complete LAHENet	27	-	-	-	-	5.8	8	8.97

Notes: NO-FS: Feature Selection Disabled.

complete LAHENet framework to clearly distinguish the impact of feature selection and secure computing protocols on the model performance at matched dimensionalities. Additionally, the accuracy variations under the classification task are further depicted in Fig. 10.



(a) Plaintext Baseline

(b) Plaintext Selected

(c) LAHENet (NO-FS)

(d) Complete LAHENet

Fig. 10. Confusion matrices under different models with SWaT.

In the SWaT anomaly detection task, feature selection reduced the dataset from 51 to 25 dimensions, resulting in only a 0.5% decrease in the F1 score. At the same 51-dimensional setting, LAHENet (NO-FS) remains within 0.13% of the Plaintext Baseline on accuracy, indicating that encrypted inference introduces only a minor and budgeted drift due to quantization. After applying secure inference, the accuracy remained at 96%, which reflects that the initial accuracy decrease is a deliberate trade-off between information retention and computational feasibility. Moreover, Fig. 10b to Fig. 10d confirm that the security inference in our framework causes almost no loss to the inference results themselves. A similar trend is observed in the regression task, where the MAPE increases by just 0.2% after feature selection and an additional 0.5% after secure inference.

These results show that the feature selection process of LAHENet achieves a reasonable balance between model accuracy and the computational demands of the edge environment. Although the end-to-end latency increases to 8 s, it still satisfies the second-level response times required for typical industrial applications such as predictive maintenance or anomaly monitoring. Moreover, the energy consumption at the ICN remains well within the practical limits for industrial sites.

D. Comparative Evaluation of LAHENet

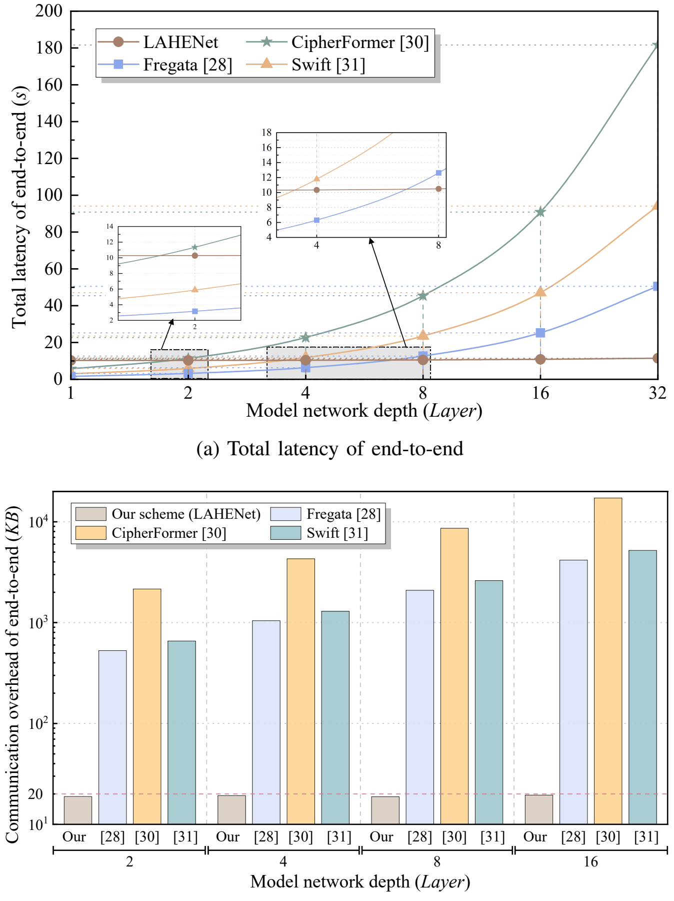
Privacy inference schemes in the IIoT edge require efficient computation, low communication overhead, and comprehensive security. We compare LAHENet with recent representative privacy-preserving inference schemes ([28], [30], [31]) and classical schemes ([34], [38]) to evaluate its applicability and advantages in the IIoT edge.

We summarize these schemes, including security mechanisms, number of interaction rounds, security properties, and reliance on hardware, in Table III. Most schemes [28], [30], [31], [34] are built on secure mechanisms such as GC and secret share (SS), where the number of interaction rounds typically increases linearly with model depth. Although they are effective in secure computation, these methods incur heavy communication costs. Additionally, the hardware dependency of Cheetah [34] and THE-V [38] limits their deployment in existing IIoT environments. In contrast, LAHENet is specifically designed for the IIoT edge topology. It requires only two constant interaction rounds and provides broader security guarantees, including forward secrecy and verifiable computation, without relying on dedicated hardware. This design

TABLE III
COMPARISON OF REPRESENTATIVE PRIVACY-PRESERVING INFERENCE SCHEMES AND THE PROPOSED LAHENET

Scheme	Security Mechanism	Interaction Number	Security Assumption	Security Properties				Hardware Reliance
				Conf.	UF	FS	VCC	
Cheetah (2021) [34]	BFV+GC	2$\Re$	RLWE	✓	–	–	–	✓
THE-V (2023) [38]	BFV+TEE	2$\Re$	RLWE+TEE	✓	✓	–	✓	✓
Fregata (2024) [28]	BFV+ASS	2$\Re$	RLWE+SS	✓	✓	–	–	–
Cipherformer (2024) [30]	BFV+GC+SS	4$\Re$	RLWE+SS	✓	✓	–	–	–
Swift (2025) [31]	CKKS+ASS	4$\Re$	RLWE	✓	–	–	–	–
Our scheme (LAHENet)	Paillier AHE+PUF	2	DCR+PUF	✓	✓	✓	✓	–

Notes: Conf.: Confidentiality; UF: Unforgeability; FS: Forward Security; VCC: Verifiable Computational Correctness; $\Re$: One round interaction.



(a) Total latency of end-to-end

(b) Communication overhead of end-to-end

Fig. 11. Comparison of the schemes in different network depths.

makes it inherently better suited to the unreliable and resource-constrained networks typical of industrial environments.

To quantify these architectural differences, we compared the performance of LAHENet with those of three hardware-independent schemes ([28], [30], and [31]). The end-to-end latency and communication overhead at different network depths are depicted in Fig. 11. With respect to shallow models ($Layer \leq 4$), schemes such as Fregata [28] exhibit low initial latency because of their ciphertext packing methods, such as BFV and CKKS. However, their mandatory multiround protocols cause latency to increase sharply as the model depth increases ($Layer > 8$), whereas LAHENet shows a superior performance trend on deeper models through its layer-folding optimization. It has a decisive advantage for predictive maintenance or advanced anomaly detection tasks, which increasingly require deeper and more complex neural networks. Because the LAHENet framework is not optimized for trivial tasks but rather for deploying long-running, complex models in the IIoT. This advantage is more obvious in Fig. 11b, where LAHENet maintains a constant communication overhead of under 20 KB. Its magnitude is lower than the millions of

TABLE IV
COMPARISON OF THE ACCURACY OF DIFFERENT SCHEMES IN ANOMALY DETECTION TASKS

Model	Accuracy	Precision	Recall	F1 score
Plaintext	99.27%	96.5%	97.5%	97%
[28]	99.20%	96.21%	97.25%	96.73%
[30]	98.67%	94.02%	95.13%	94.57%
[31]	99.16%	95.95%	97.18%	96.56%
LAHENet	99.18%	96.1%	97.2%	96.65%

bytes and linearly increasing costs of other solutions, making it uniquely feasible in bandwidth-constrained environments.

We compare the effects of Fregata, Swift, CipherFormer, and LAHENet on model accuracy on the SWaT dataset. As shown in Table IV, all the schemes have lower accuracy than the plaintext baseline does. However, LAHENet achieves an accuracy close to that of the plaintext baseline and remains competitive with Fregata [28] (99.20%) and Swift [31] (99.16%) while significantly outperforming CipherFormer [30] (98.67%). These results show that the security and performance advantages of LAHENet are not achieved at the expense of model accuracy.

Overall, LAHENet provides a well-balanced solution, which is especially suited for IIoT environments that prioritize comprehensive security, efficient resource utilization, and low-latency responsiveness.

VII. Conclusion

In this paper, we proposed the LAHENet framework, which is the first end-to-end privacy-preserving and real-time private inference framework for the IIoT edge environment. It uses an innovative additive homomorphic signcryption protocol combined with PUF-based key management and a verifiable ciphertext-domain inference mechanism. These designs achieve efficient ciphertext processing and minimal overhead while ensuring full-lifecycle data protection in resource-constrained IIoT environments. The results of performance experiments show that LAHENet maintains end-to-end inference latency within milliseconds and maintains communication overhead under 30 KB per inference. An accuracy experiment on the SWaT dataset demonstrates that our scheme differs from plaintext inference by less than 1%, which is better than existing private inference schemes. Moreover, we analysed the robustness and security of the framework under ROM. These results indicate the strong potential of LAHENet for practical deployment in industrial edge environments.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (2024YFB3108200).

References

[1] R. Tallat, A. Hawbani, X. Wang, A. Al-Dubai, L. Zhao, Z. Liu, G. Min, A. Y. Zomaya, and S. H. Alsamhi, "Navigating industry 5.0: A survey of key enabling technologies, trends, challenges, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 1080–1126, 2023.

[2] M. Sharma, A. Tomar, and A. Hazra, "Edge computing for industry 5.0: Fundamental, applications, and research challenges," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19070–19093, 2024.

[3] M. A. Jarwar, J. W. C. FREng, and S. Ali, "Modelling industrial iot security using ontologies: a systematic review," *IEEE Open Journal of the Communications Society*, 2025.

[4] Z. A. Shaikh, F. Hajjej, Y. D. Uslu, S. Yüksel, H. Dınçer, R. Alroobaea, A. M. Baqasah, and U. Chinta, "A new trend in cryptographic information security for industry 5.0: A systematic review," *IEEE Access*, vol. 12, pp. 7156–7169, 2024.

[5] P. Kumar, R. Kumar, G. P. Gupta, R. Tripathi, and G. Srivastava, "P2tif: A blockchain and deep learning framework for privacy-preserved threat intelligence in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6358–6367, 2022.

[6] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning*, pp. 201–210, 2016.

[7] W. Ao and V. N. Boddeti, "AutoFHE: Automated adaption of CNNs for efficient evaluation over FHE," in *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 2173–2190, 2024.

[8] J. Lin, Y. Miao, L. Wei, T. Leng, and K.-K. R. Choo, "Efficient secure inference scheme in multiparty settings for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 10, pp. 11877–11886, 2024.

[9] R. Wu, B. Wang, and Z. Zhao, "Epbnn: Efficient and private inference for binary neural network," *IEEE Internet of Things Journal*, 2025.

[10] B. Deebak and S. O. Hwang, "Privacy-preserving learning model using lightweight encryption for visual sensing industrial iot devices," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2025.

[11] S. A. Khowaja, K. Dev, N. M. F. Qureshi, P. Khuwaja, and L. Foschini, "Toward industrial private ai: A two-tier framework for data and model security," *IEEE Wireless Communications*, vol. 29, no. 2, pp. 76–83, 2022.

[12] I. B. Ababio, J. Bieniek, M. Rahouti, T. Hayajneh, M. Aledhari, D. C. Verma, and A. Chehri, "A blockchain-assisted federated learning framework for secure and self-optimizing digital twins in industrial iot," *Future Internet*, vol. 17, no. 1, p. 13, 2025.

[13] E. Chou, J. Beal, D. Levy, S. Yeung, A. Haque, and L. Fei-Fei, "Faster cryptonets: Leveraging sparsity for real-world encrypted inference," *arXiv preprint arXiv:1811.09953*, 2018.

[14] T. Chen, H. Bao, S. Huang, L. Dong, B. Jiao, D. Jiang, H. Zhou, J. Li, and F. Wei, "The-x: Privacy-preserving transformer inference with homomorphic encryption," *arXiv preprint arXiv:2206.00216*, 2022.

[15] Q. Lou and L. Jiang, "She: A fast and accurate deep neural network for encrypted data," *Advances in neural information processing systems*, vol. 32, 2019.

[16] M. R. Abou Harb and B. Celiktas, "Privacy-preserving machine learning: Ann activation function estimators for homomorphic encrypted inference," *IEEE Access*, 2025.

[17] D. Kim and C. Guyot, "Optimized privacy-preserving cnn inference with fully homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2175–2187, 2023.

[18] W. Wu, J. Wang, Y. Zhang, Z. Liu, L. Zhou, and X. Lin, "Vpip: Values packing in paillier for communication efficient oblivious linear computations," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4214–4228, 2023.

[19] L. Liu, X. Du, M. Ma, and D. Wang, "A privacy protection strategy based on homomorphic encryption and neural networks," in *Proceedings of the 2024 8th International Conference on Electronic Information Technology and Computer Engineering*, pp. 586–591, 2024.

[20] J. Moon, D. Yoo, X. Jiang, and M. Kim, "Thor: Secure transformer inference with homomorphic encryption," *Cryptology ePrint Archive*, 2024.

[21] J. Zhang, X. Yang, L. He, K. Chen, W.-j. Lu, Y. Wang, X. Hou, J. Liu, K. Ren, and X. Yang, "Secure transformer inference made non-interactive," *Cryptology ePrint Archive*, 2024.

[22] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via minionn transformations," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 619–631, 2017.

[23] M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar, "XONN: XNOR-based oblivious deep neural network inference," in *28th USENIX Security Symposium (USENIX Security 19)*, pp. 1501–1518, 2019.

[24] F. Boemer, A. Costache, R. Cammarota, and C. Wierzynski, "ngraph-he2: A high-throughput framework for neural network inference on encrypted data," in *Proceedings of the 7th ACM workshop on encrypted computing & applied homomorphic cryptography*, pp. 45–56, 2019.

[25] M. Hao, H. Li, H. Chen, P. Xing, and T. Zhang, "Fastsecnet: An efficient cryptographic framework for private neural network inference," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2569–2582, 2023.

[26] K. Cheng, N. Xi, X. Liu, X. Zhu, H. Gao, Z. Zhang, and Y. Shen, "Private inference for deep neural networks: A secure, adaptive, and efficient realization," *IEEE Transactions on Computers*, vol. 72, no. 12, pp. 3519–3531, 2023.

[27] Z. Hu, L. Chen, Y. Wang, and P. Zhang, "A secure convolutional neural network inference model based on homomorphic encryption," in *2024 7th World Conference on Computing and Communication Technologies (WCCCT)*, pp. 17–23, 2024.

[28] X. Yang, J. Chen, Y. Li, K. He, X. Huang, Z. Jiang, H. Bai, and R. Du, "Fregata: fast private inference with unified secure two-party protocols," *IEEE Transactions on Information Forensics and Security*, 2024.

[29] M. Zheng, Q. Lou, and L. Jiang, "Primer: Fast private transformer inference on encrypted data," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2023.

[30] W. Wang and Y. Kuang, "Cipherformer: Efficient transformer private inference with low round complexity," in *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 3054–3059, 2024.

[31] Y. Fu, Y. Tong, Y. Ning, T. Xu, M. Li, J. Lin, and D. Feng, "Swift: Fast secure neural network inference with fully homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, 2025.

[32] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *27th USENIX security symposium (USENIX security 18)*, pp. 1651–1669, 2018.

[33] W. Z. Srinivasan, P. Akshayaram, and P. R. Ada, "Delphi: A cryptographic inference service for neural networks," in *Proc. 29th USENIX secur. symp*, vol. 3, 2019.

[34] B. Reagen, W.-S. Choi, Y. Ko, V. T. Lee, H.-H. S. Lee, G.-Y. Wei, and D. Brooks, "Cheetah: Optimizing and accelerating homomorphic encryption for private inference," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 26–39, 2021.

[35] S. Singh, S. Singh, S. Gudaparthi, X. Fan, and R. Balasubramonian, "Hyena: balancing packing, reuse, and rotations for encrypted inference," in *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 3091–3108, 2024.

[36] K. Rajasekar, R. Loh, K. W. Fok, and V. L. Thing, "Privacy preserving layer partitioning for deep neural network models," in *2024 IEEE Conference on Artificial Intelligence (CAI)*, pp. 1129–1135, 2024.

[37] Z. Liu, Y. Luo, S. Duan, T. Zhou, and X. Xu, "Mirrornet: A tee-friendly framework for secure on-device dnn inference," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–9, 2023.

[38] Y. Wei, X. Wang, S. Bian, W. Zhao, and Y. Jin, "The-v: Verifiable privacy-preserving neural network via trusted homomorphic execution," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–9, 2023.

[39] W. Xu, H. Zhu, Y. Zheng, F. Wang, J. Hua, D. Feng, and H. Li, "Tonn: An oblivious neural network prediction scheme with semi-honest tee," *IEEE Transactions on Information Forensics and Security*, 2024.

[40] I. Ullah, N. Ul Amin, M. Zareei, A. Zeb, H. Khattak, A. Khan, and S. Goudarzi, "A lightweight and provable secured certificateless signcryption approach for crowdsourced iiot applications," *Symmetry*, vol. 11, no. 11, p. 1386, 2019.

[41] I. Ullah, A. Alomari, A. M. Abdullah, N. Kumar, A. Alsirhani, F. Noor, S. Hussain, and M. A. Khan, "Certificate-based signcryption scheme for securing wireless communication in industrial internet of things," *IEEE Access*, vol. 10, pp. 105182–105194, 2022.

[42] T.-H. Kim, G. Kumar, R. Saha, W. J. Buchanan, T. Devgun, and R. Thomas, "Lisp-xk: extended light-weight signcryption for iot in resource-constrained environments," *IEEE Access*, vol. 9, pp. 100972–100980, 2021.

[43] G. S. Rao, G. Thumbur, R. B. Amarapu, G. N. Bhagya, and P. V. Reddy, "A new lightweight and secure certificateless aggregate signcryption scheme for industrial internet of things," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 10563–10574, 2023.

[44] M. Barbareschi, V. Casola, A. De Benedictis, E. La Montagna, and N. Mazzocca, "On the adoption of physically unclonable functions to secure iiot devices," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7781–7790, 2021.

[45] Q. Zhang, J. Wu, H. Zhong, D. He, and J. Cui, "Efficient anonymous authentication based on physically unclonable function in industrial internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 233–247, 2022.

[46] C.-I. Fan, C.-I. Lai, and D. V. Medhane, "Cake-puf: a collaborative authentication and key exchange protocol based on physically unclonable functions for industrial internet of things," *IEEE Internet of Things Journal*, 2024.

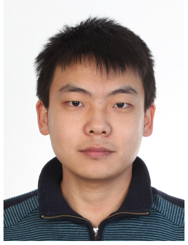
[47] L. Yu, W. Wu, and L. Mei, "A lightweight cross-layer mutual authentication with key agreement protocol for iiot," *IEEE Internet of Things Journal*, 2024.

[48] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*, pp. 223–238, 1999.

[49] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *International conference on critical information infrastructures security*, pp. 88–99, 2016.

Mowei Gong pursuing her Ph.D degree in Beijing University of Technology, Beijing China. She received her M.S. degree from the Beijing University of Technology, Beijing, China. Her research interests include Industrial Internet of Things security, privacy data protection, and cryptography.

Zhe Li pursuing his Ph.D degree in Beijing University of Technology, Beijing China. He received his M.S. degree from the Chemnitz University of Technology, Chemnitz, Germany. His research interests include Internet of Vehicles security, identity authentication, and cryptography.



Xuepeng Lu received the B.S. degree from China University of Petroleum (Beijing) in 2009. He is currently Vice President at Beijing Luo'an Technology Co., Ltd., where he focuses on ICS cybersecurity across the power, petroleum, and manufacturing sectors. He has led major international projects, including the successful completion of the cybersecurity project for the power monitoring system at Cambodia's Lower Sesan 2 Hydropower Station (2020) and the design and implementation of industrial control security systems for Iraq's Majnoon Oil Field (2024). His research interests are centered on IIoT security, including industrial protocol reverse engineering, PLC/RTU firmware security, intrusion detection, and vulnerability discovery.

Bei Gong (Member, IEEE) received a B.S. degree from Shandong University in 2005 and a Ph.D. degree from the Beijing University of Technology in 2012. In the past five years, he has published more than 30 papers in first-class SCI/EI and other international famous journals and top international conferences in relevant research fields. His research interests include trusted computing, Internet of Things security, Industrial Internet of Things, and privacy data protection.