Engineering Applications of Artificial Intelligence 135 (2024) 108862

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Enhanced collaborative intrusion detection for industrial cyber-physical systems using permissioned blockchain and decentralized federated learning networks

Junwei Liang [a], Muhammad Sadiq [a,*], Geng Yang [a], Kai Jiang [a], Tie Cai [a], Maode Ma [b]

[a] Shenzhen Institute of Information Technology, Shenzhen, China
[b] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Industrial cyber-physical systems
Intrusion detection systems
External classifier-generative adversarial network
Local differential privacy
Decentralized federated distillation
Hyperledger fabric

ABSTRACT

In the digital economy, the security of Cyber-Physical Systems (CPSs) is paramount. Despite the deployment of various Intrusion Detection Systems (IDSs) in industrial CPSs, the primary obstacles to the advanced research include class imbalance, privacy gap, model homogenization, and arbitrary aggregation. Thus, in this paper, a Permissioned Blockchain-enabled decentralized federated distillation Generative Adversarial Network (GAN), namely PB-fdGAN, is proposed for Collaborative IDSs (CIDSs) in industrial CPSs. The novel Local Differential Privacy (LDP)-external classifier GAN (ecGAN) addresses class imbalance and privacy concerns by integrating label conditions into the latent space and using Wasserstein distance for stability in semi-supervised learning. Additionally, a Decentralized Federated Distillation (DFD) scheme allows for collaborative model building without data exchange, enhancing privacy and diversity. Moreover, a Quality of Service (QoS)-consortium blockchain framework with a new QoS evaluation strategy ensures reliable and effective model aggregation. The experimental evaluation demonstrates the high effectiveness of PB-fdGAN in detecting various types of cyber threats and the superiorities over state-of-the-art IDS solutions.

1. Introduction

The burgeoning digital economy critically relies on the secure and efficient operation of industrial Cyber-Physical Systems (CPSs), which represent a synergy of computing, control, communication, and physical processes. As foundational components in the rapidly expanding realms of smart grids, autonomous transportation systems, and unmanned factories, these systems are pivotal in advancing research and development (Pivoto et al., 2021). They are not only essential for modern industrial operations but also act as key enablers in the digital transformation of economies, integrating traditional sectors with innovative technologies to enhance productivity and economic growth (Mourtzis, 2023). Industrial CPSs, characterized by their large-scale, distributed nature, embody the cooperative and heterogeneous spirit of the Internet-of-Things (IoTs) within the industrial sector. By integrating traditional industrial infrastructures with cutting-edge technologies such as 5G, software-defined networking, and artificial intelligence, they facilitate an omnipresent perception of the environment, embedded computing, and networked communication and control (Cyber-Physical System, 2023). The robust architecture of an industrial CPS, depicted in Fig. 1, consists of three primary layers: the physical layer, equipped with numerous sensors and actuators for environmental data acquisition and infrastructure interaction; the control layer, which processes this data and provides visual feedback to operators; and the network layer, acting as the connective tissue between the cyber and physical realms to ensure the seamless flow of information. In an era where economic activities are increasingly digitized, the security of these systems is not just a technical concern but a vital economic imperative. Strengthening the resilience of industrial CPSs against cyber threats is crucial for safeguarding the digital economy, ensuring the continuity of industrial operations, and maintaining the trust that fuels digital commerce and innovation. As academia and industry collaborate to enhance the capabilities of sensors and actuators with advanced communication and data processing functions, ensuring the cybersecurity of industrial CPSs is paramount for the evolution and prosperity of the digital economic landscape (Salau et al., 2022).

Unsurprisingly, the enormous usefulness of industrial CPSs in transportation networks, energy systems, and water/gas distribution

* Corresponding author.
E-mail address: sadiq@sziit.edu.cn (M. Sadiq).

https://doi.org/10.1016/j.engappai.2024.108862
Received 13 December 2023; Received in revised form 8 June 2024; Accepted 13 June 2024
Available online 22 June 2024

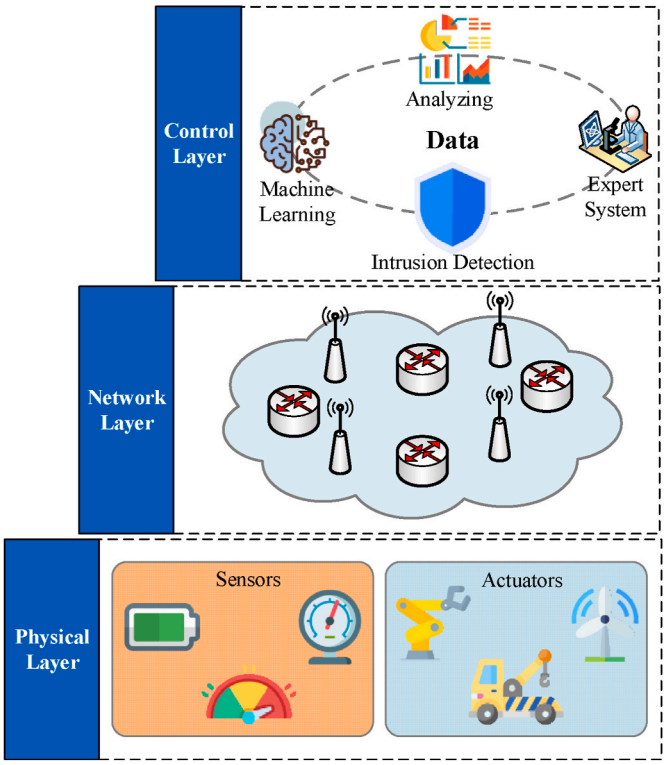

Fig. 1. Generic architecture of industrial CPS.

networks has been stimulating adversaries to launch attacks on the systems, and the rapid fusion of advanced networking and computing technologies has dramatically expanded the threat landscape leaving numerous potential vulnerabilities unattended. For instance, notable incidents include the ransomware assault on Colonial Pipeline, which led to a significant disruption in fuel distribution across the Eastern United States, highlighting the imperative for robust cybersecurity defenses in transportation networks (George et al., 2024). Another case in point is the cyberattack on a water treatment facility in Florida, where the potential for adversaries to manipulate chemical levels in the water supply underscored the dire consequences of inadequate security measures (Cervini et al., 2022). The deployment of Intrusion Detection Systems (IDSs) is one of the most important approaches to protect industrial CPSs against various threats, as it has the ability to detect both internal and external attacks with high accuracy (Liang et al., 2020), (Liang et al., 2021). In recent years, many studies have been conducted for IDSs in industrial CPSs, focusing on the dual challenges of improving IDS capabilities and fostering collaboration across systems for enhanced security. This exploration has led to the enhancement of IDS frameworks through the integration of signature-based and anomaly-based analyses, alongside the adoption of avant-garde computational models aimed at elevating the precision and efficiency of anomaly detection (Quincozes et al., 2021)-(Keshk et al., 2019). Furthermore, research has explored federated learning and blockchain technology to promote collaborative IDS efforts and trust among systems (Belenguer et al., 2022)-(Aloqaily et al., 2022). These innovative approaches signify a concerted effort within the cybersecurity community to develop a more cohesive and resilient defense mechanism against the evolving spectrum of cyber threats targeting industrial infrastructures.

Even though various IDSs have been deployed in industrial CPSs, most advanced works typically rely on limited and imbalanced datasets, which are confined to individual industrial systems, failing to capitalize on a broader, distributed data pool across various CPSs. The core bottlenecks are attributable to unresolved challenges: i) Class imbalance commonly exists in the well-known datasets, which significantly hindered the effectiveness of detecting anomalies; ii) Few efforts have been made for privacy preservation gap, resulting in data becoming siloed and leading to isolated data islands that cannot benefit from collective insights; iii) The merging of homogeneous models trained on heterogeneous datasets without nuanced integration strategies leads to diminished detection capabilities. These critical challenges significantly impede the efficacy of IDSs in industrial CPSs, yet there has been a noticeable scarcity of initiatives aimed at mitigating these challenges. To address these issues, a Permissioned Blockchain-enabled decentralized federated distillation Generative Adversarial Network (GAN), called PB-fdGAN, is proposed. The advanced technologies in PB-fdGAN for industrial CPSs not only strengthens their security but also promotes a more interconnected and robust digital economy. By ensuring the reliability and security of the digital infrastructure, PB-fdGAN plays a pivotal role in fostering trust and enabling the seamless operation of digital services, which are the backbone of modern economic systems. The major contributions of PB-fdGAN are as follows:

- To deal with the restricted, imbalanced datasets of industrial CPSs, a collaboratively adapted Local Differential Privacy (LDP)-external classifier GAN (ecGAN) intrusion detection model is proposed for semi-supervised classification, which is the first to develop External Classifier (EC)-GAN with employing Wasserstein distance to provide a smooth measure for stabilizing the gradient descent process and conducting label condition as an extension into the latent space to improve the classification performance. For preventing privacy leakage in collaboration (or data island), the LDP technique is leveraged by adding well-designed noise into the gradients during the training process of LDP-ecGAN.
- For overcoming model homogenization, a newly-designed Decentralized Federated Distillation (DFD) collaboration scheme with advanced compatibility and interoperability is presented for LDP-ecGAN, which enables multiple CPSs collectively to build a comprehensive intrusion detection model without directly exchanging their sub-network flows and an isomorphic neural network model. For the DFD collaboration scheme, the generator is jointly trained by dispatching its generated data and leveraging the feedback losses from the distributed discriminators for augmenting the generation capability, while the classifier and discriminator are updated based on the decentralized federated distillation protocol that transfers knowledge among co-modelling participants without sharing a template model for general applicability.
- To combat the challenges of arbitrary aggregation, Quality of Service (QoS)-Hyperledger Fabric (HLF) framework is crafted, which incorporates an innovative QoS evaluation methodology to quantitatively select the most effective intrusion detection models based on certain criteria and performance metrics, favoring a performance-weighted aggregation approach over the random amalgamation of all candidate models. Additionally, the DFD collaboration scheme is modeled as a flexible QoS mode in the QoS-HLF framework with using Chain Codes (CCs) to ensure the automatic and reliable execution of collaboration as agreed upon by all co-modelling participants for defending against poisoning and membership inference attacks.

The remainder of this paper is organized as follows. Section 2 delves into related works, offering context and insights. In Section 3, the background and preliminaries to this paper are provided. The collaboration of PB-fdGAN, including LDP-ecGAN and the DFD collaboration scheme, is explored in Sections 4. The newly designed QoS-HLF framework is presented in Section 5. Section 6 demonstrates the experimental results and theoretical analysis in detail. At last, Section 7 encapsulates the findings and draws conclusions from this paper.

2. Related works

Ensuring the security of industrial CPSs has emerged as a critical concern, drawing considerable attention from the research community in recent year. A pivotal challenge in this area is the presence of malicious activities that can severely compromise the integrity and reliability of these systems. In response, a significant volume of research has been dedicated to developing IDSs tailored for industrial CPSs. These efforts have primarily concentrated on two key branches as follows:

2.1. Improvement and application of IDSs

i) Universal IDSs: In (Quincozes et al., 2021)-(Khan et al., 2022), signature-based analysis has been presented to check the nodes in industrial CPSs for compliance according to pre-defined rule sets. Besides, anomaly-based analysis is further applied in (Althobaiti et al., 2021)-(Hao et al., 2021) by employing statistical IDS solutions to recognize the intrusions that are not yet covered by preset rules. These IDS solutions are not tailored to any specific attack types, such that a complex analysis of observed data is expected. It generates a large number of false alarms with flagging normal traffic as malicious, and sophisticated or complex attacks that are specifically crafted can easily evade being identified as malicious. **ii)** Custom IDSs: In (Wang et al., 2022a), a knowledge distillation model based on triplet convolution neural network is proposed to improve the model performance and greatly enhance the speed of anomaly detection as well as reduce the complexity of the model for industrial CPSs. A cognitive computing-based intrusion detection method named border-line SMOTE is proposed in (Gao et al., 2022). By using the specification rules and a lightweight neural network, the communication overhead of IDSs is reduced. In (Alohali et al., 2022), a weighted voting-based ensemble model is developed with using recurrent neural network (RNN), bi-directional long short-term memory (Bi-LSTM), and deep belief network (DBN) for intrusion detection. Although IDSs in (Quincozes et al., 2021)-(Alohali et al., 2022) are proved in detecting normal and abnormal traffic with high accuracy and efficiency, the datasets adopted in the literature are only appropriate for IDSs in the early stage, as the network traffic in these datasets are classic and widespread but lack of specialization. Most importantly, class-imbalanced problem commonly exists in the well-known datasets of industrial CPSs, thereby the most cutting-edge IDSs face a bottleneck in effectively detecting the anomalies with sparse training samples. **iii)** Privacy IDSs: In (Khan et al., 2021), a privacy-conserving intrusion detection framework is proposed, employing data pre-processing for privacy and using a particle swarm optimization-based probabilistic neural network for effective detection of cyber-attacks. Another innovative approach is presented in (Keshk et al., 2019) for privacy-preserving anomaly detection in CPSs, by integrating a data pre-processing module for privacy preservation with a Gaussian Mixture Model (GMM) and Kalman Filter (KF) based anomaly detection module. Privacy-preserving is essential to IDSs in industrial CPSs for safeguarding sensitive data and ensuring compliance with privacy regulations, while effectively detecting and mitigating cyber threats. Despite its importance, the pursuit of comprehensive privacy-preserving solutions for IDSs in industrial CPSs has seen limited advancement, and the in-depth exploitation of privacy preservation remains an open problem.

2.2. Synergistic approaches and collaborative IDSs

i) Federated Learning: In recent years, Federated Learning (FL)-based solutions have been widely implemented in IDSs for industrial CPSs (Belenguer et al., 2022). In (Tahir et al., 2021), a deep-federated learning-based decentralized detection method using an attentive aggregation is exploited, which is capable of parallel computing and can reliably identify the stealthy false data injection attacks on all the nodes simultaneously. Similarly, a federated deep learning scheme, named DeepFed, is proposed to detect cyber threats against industrial CPSs in (Li et al., 2020a), and a general global detection model, called EEFED, is designed for collaboratively improving the performance of a single local model against cyberattacks in (Huang et al., 2022). The most prominent issue faced by the state-of-the-art FL-based IDSs is model homogenization (Li et al., 2020b). Employing the same IDS model across all CPSs leads to a loss of model diversity, poor model adaptability, and limited customization. **ii)** Blockchain Technology: In addition to FL-based solutions, blockchain and distribute ledger technology have been adopted to achieve the trust among Collaborative IDSs (CIDSs) in industry 4.0 CPSs and industrial IoT systems in (Rahman et al., 2022) and (Kumar et al., 2021) respectively. The authors claim that the solutions can enhance the credibility of CIDSs by enabling data accountability to prevent against various inference attacks. However, the public blockchains the CIDSs employ in (Rahman et al., 2022)-(Kumar et al., 2021) ineluctably post several critical challenges, including the participation of adversaries in uploading and auditing sensitive information, the high electrical resource requirements for mining blocks and reaching network consensus, and the anonymity of validators that increases the risk of collisions and 51% attack. Instead of using public blockchain, permissioned blockchain-based hierarchical CIDS solutions that maintains quick, secure, and accurate decision-making for industrial CPSs are proposed in (Mansour, 2022)-(Aloqaily et al., 2022). Unfortunately, the current CIDSs based on federated learning and blockchain technology normally experience a degradation in detection performance following arbitrary multi-model aggregation. This can be attributed to the heterogeneous nature of datasets across different industrial CPSs, resulting in inconsistent local and global optima.

3. Background and preliminaries

In this section, the concepts considered in our work, i.e., GAN, Wasserstein GAN (WGAN), EC-GAN, and DP are introduced in brief. The mathematical notations and important parameters frequently used in this paper are summarized in Table 1.

3.1. GAN, WGAN, and EC-GAN

GAN is a generative neural network that can learn the mapping from a simple latent (or noise) distribution to an arbitrarily complex data

Table 1
Summary of notations and parameters.

Symbol	Description
$n \in N, k \in K$	Numbers of all and candidate providers
$u, \{v\}_k^K$	Cooperative service requester and providers
$\mathscr{G}, \mathscr{D}, \mathscr{C}$	GAN modules of service requester
ϕ, ω, θ	GAN modules' weights of service requester
$(\mathscr{G}/\mathscr{D}/\mathscr{C})_{n/k}$	GAN modules of n/k-*th* service provider
$(\phi/\omega/\theta)_{n/k}$	GAN modules' weights of n/k-*th* provider
$b \in B$	Batch size of dataset sampling
$\{s^b\}_b^B \sim p(S)$	Data samples from requester's dataset
$s^b = (x^b, y^b)$	A data sample composed of data and a label
$z \sim p(z)$	Random noise for GAN's generators
$\{l^b\}_b^B = L$	Randomly generated labels
$\widehat{x} = \mathscr{G}(x^b \mid l^b)$	Synthetic data of requester's generator
$T_\phi, T_\omega, T_\theta, T$	Iterations of local training/global training
$\{e_{n,t}^{(b)}\}_b^B$	Error terms of n provider at t epoch
$F_{(\theta/\omega)_n,t}^{\{m\}/0/1}$	Local logit vectors of n provider at t epoch
$X_{test}, Y_{label}/Y_{RoF}$	Hybrid testing data and respective labels

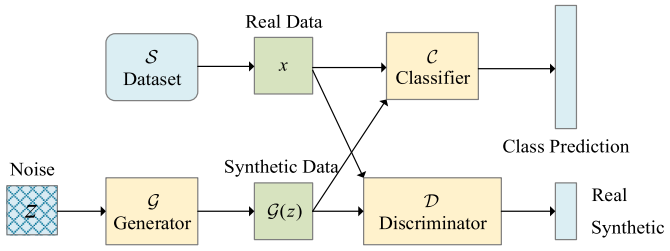

Fig. 2. External classifier GAN model (Haque, 2020).

distribution to generate realistic data (Gui et al., 2021). Two models in GAN are trained simultaneously, in which a generator $\mathscr{G}$ attempts to generate the synthetic samples approximating the real data distribution, and a discriminator $\mathscr{D}$ estimates the probability that a sample comes from the real dataset rather than the outputs of $\mathscr{G}$. Let $p(z)$ be the input noise distribution of $\mathscr{G}$ and $p(x)$ be the real data distribution. GAN aims at training $\mathscr{G}$ and $\mathscr{D}$ to play the two-player minimax game with value function $\mathscr{V}(\mathscr{G},\mathscr{D})$ as Eq. (1):

$$\min_{\mathscr{G}}\max_{\mathscr{D}}\mathscr{V}(\mathscr{G},\mathscr{D})=\mathbb{E}_{x\sim p(x)}[\log(\mathscr{D}(x))]+\mathbb{E}_{z\sim p(z)}[\log(1-\mathscr{D}(\mathscr{G}(z)))] \quad (1)$$

To overcome the mode collapse and gradient loss problems of the original GAN (Gui et al., 2021), WGAN (Wasserstein GAN, 2023) upgrades GAN by using the Wasserstein distance as the objective function instead of the Jensen-Shannon Divergence. It solves a different two-player minimax game given by Eq. (2):

$$\min_{\mathscr{G}}\max_{\omega}\mathbb{E}_{x\sim p(x)}[f_\omega(x)]-\mathbb{E}_{z\sim p(z)}[f_\omega(\mathscr{G}(z))] \quad (2)$$

where ω represents the discriminator parameters or weights of WGAN, and $\{f_\omega(x)\}_\omega$ are a parameterized family of functions that are all K-Lipschitz with respect to x for a Lipschitz constant K.

EC-GAN is a variant generative model that attaches an external classifier to the original GAN to improve classification performance in restricted, fully-supervised datasets (Haque, 2020). The model is composed of three separate models, i.e., a generator $\mathscr{G}$, a discriminator $\mathscr{D}$, and a classifier $\mathscr{C}$, as shown in Fig. 2. At every training iteration, $\mathscr{G}$ is given a noise vector z to generate the corresponding output $\mathscr{G}(z)$, while $\mathscr{D}$ is then updated to better distinguish between x and $\mathscr{G}(z)$, and the losses for $\mathscr{D}$ and $\mathscr{G}$ are defined as Eqs. (3) and (4):

$$\mathscr{L}_{\mathscr{D}}(x,z)=\mathscr{L}_{bce}(\mathscr{D}(x),1)+\mathscr{L}_{bce}(\mathscr{D}(\mathscr{G}(z)),0) \quad (3)$$

$$\mathscr{L}_{\mathscr{G}}(z)=\mathscr{L}_{bce}(\mathscr{D}(\mathscr{G}(x)),1) \quad (4)$$

where $\mathscr{L}_{bce}(\bullet)$ is binary cross-entropy loss. The classifier $\mathscr{C}$ is trained iteratively in parallel with $\mathscr{G}$ in a standard fashion on a real data x and its corresponding label y before utilizing the generated outputs $\{\mathscr{G}(z)\}_z$ for the supplementary training, which is the semi-supervised portion of EC-GAN as $\{\mathscr{G}(z)\}_z$ have no associated labels. To create the respective labels for $\{\mathscr{G}(z)\}_z$, a pseudo-labeling scheme is adopted in EC-GAN that assumes a label based on the most likely class predicted by the current state of the classifier $\mathscr{C}$. A generated $\mathscr{G}(z)$ and the corresponding label are only retained if the model $\mathscr{C}$ determines the class of the generated $\mathscr{G}(z)$ with a probability above the pseudo-labeling threshold τ as $\text{argmax}(\mathscr{C}(\mathscr{G}(z)))>\tau)$. The classifier $\mathscr{C}$'s loss of EC-GAN can be calculated by Eq. (5):

$$\mathscr{L}_{\mathscr{C}}(x,y,z)=\mathscr{L}_{ce}(\mathscr{C}(x),y)+\lambda\mathscr{L}_{ce}(\mathscr{C}(\mathscr{G}(z)),\text{argmax}(\mathscr{C}(\mathscr{G}(z)))>\tau) \quad (5)$$

where $\mathscr{L}_{ce}(\bullet)$ is cross-entropy loss and λ is the hyperparameter that controls the relative importance of $\mathscr{G}(z)$ compared to the conjugated true data (x,y). Consequently, EC-GAN in (Haque, 2020) is demonstrated effective to improve the classification performance in low-sample, real-world datasets.

3.2. Differential privacy

DP constitutes a strong standard for guaranteeing the privacy of dataset-aggregating algorithms, which is defined in terms of the application-specific concepts of adjacent datasets (Dong et al., 2022). Let S be a sensitive dataset to be published, and DP refers to the process that S is modified using a randomized mechanism $\mathscr{M}$, such that the output O of $\mathscr{M}$ does not reveal the information about any particular tuples in S. The formal definition of DP is detailed below.

Definition: ((ϵ,δ)-differential privacy (Bu et al., 2020)). If an (randomized) algorithm $\mathscr{M}$ satisfies (ϵ,δ)-differential privacy, for any pair of neighboring datasets S and $\widetilde{S}$ differing in at most one user's one attribute value, and for all sets of possible outputs $O\subseteq Range(\mathscr{M})$, we have:

$$Pr[\mathscr{M}(S)\in O]\le exp(\epsilon)\bullet Pr[\mathscr{M}(\widetilde{S})\in O]+\delta \quad (6)$$

where $Pr[\bullet]$ denotes the probability of an event, and (ϵ,δ)-differential privacy becomes ϵ-differential privacy when $\delta=0$.

Algorithmically, a common paradigm to approximate a deterministic real-value function $f:S\to\mathbb{R}$ with a differentially private mechanism is to introduce independent Gaussian noise. This is known as the Gaussian mechanism (Zhu et al., 2020).

Lemma: (Gaussian mechanism (Zhu et al., 2020)). As the L2-sensitivity of f is determined as $\Delta f=\max_{S,\widetilde{S}}\|f(S)-f(\widetilde{S})\|_2$ over all pairs of neighboring $(S,\widetilde{S})$, the Gaussian mechanism to provide (ϵ,δ)-differential privacy is defined by:

$$\mathscr{M}(S)=f(S)+\sim\mathscr{N}(0,\sigma^2) \quad (7)$$

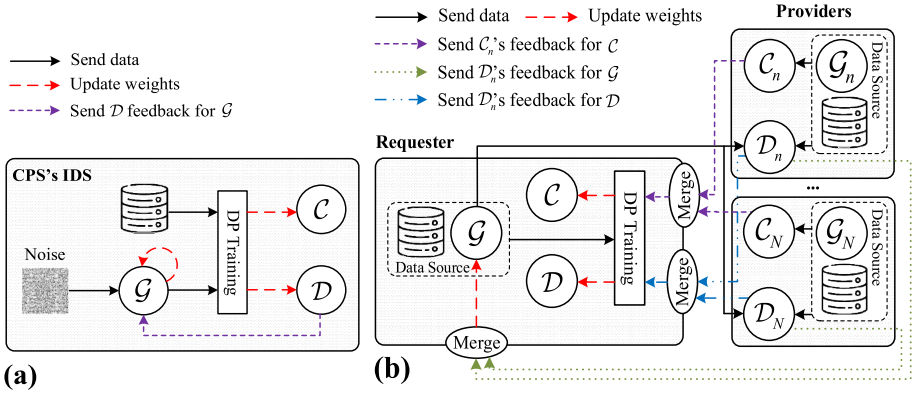

Fig. 3. Collaboration of PB-fdGAN: (a) Local Training for LDP-ecGAN; (b) Global Training via DFD collaboration scheme.

where $\mathcal{N}(0,\sigma^2)$ is the normal (Gaussian) distribution with mean 0, and σ is the standard deviation that is normally set to $\Delta f\sqrt{2\log\left(\frac{1.25}{\delta}\right)}/\epsilon$.

The Gaussian mechanism is one of the widely-adopted randomization mechanisms to achieve DP (Zhu et al., 2020). With using Eqs. (6) and (7), the Gaussian mechanism can protect the membership of a data point in a dataset by associating the output of the data point with the distribution that does not change too much if certain data are perturbed or removed.

4. Collaboration in PB-fdGAN

For collectively building a secure and private-assured CIDS over whole industrial CPSs, the collaboration of the proposed PB-fdGAN is presented, the basic idea of which is updating the newly-designed LDP-ecGAN in reference to our DFD collaboration scheme. The complete workflow of a PB-fdGAN collaboration session can be described in two processes: (i) *local training for LDP-ecGAN* inside of a random industrial CPS (Fig. 3(a)); (ii) *global training* via *the DFD collaboration scheme* across industrial CPSs (Fig. 3(b)). During the local training process, every CPS leverages its local dataset to train its own LDP-ecGAN that integrates with Wasserstein distance and label condition in a privacy-preserving way. Once completing the local training, any collaboration-oriented CPS, namely a cooperative service requester, can initiate the global training process by networking with multiple peers, i.e., its corresponding providers, to build a comprehensive IDS model based on the DFD collaboration scheme.

Particularly, in the DFD collaboration scheme (Fig. 3(b)), there is one cooperative service requester u and N corresponding providers $\{v_n\}_n^N$. The service requester and *n-th* provider v_n are equipped with a local dataset S and S_n respectively, and the entire dataset is donated by $\bigcup_{n=1}^{N} S_n \cup S$. A generator $\mathscr{G}$, a discriminator $\mathscr{D}$, and an external classifier $\mathscr{C}$ are hosted in the requester with a privacy-preserving layer, while v_n operates its own discriminator $\mathscr{D}_n$ and classifier $\mathscr{C}_n$ for peer-to-peer collaboration services. The weights of $\mathscr{G}$, $\mathscr{D}$, and $\mathscr{C}$ are ϕ, ω, and θ respectively, and locally renamed as ϕ_n, ω_n, and θ_n for the *n-th* provider. The detailed procedures of the two processes are elaborated in the following.

4.1. Local training for LDP-ecGAN

Here, the novel LDP-ecGAN model is designed for IDSs in industrial CPSs. For generalization, a generic architecture of LDP-ecGAN is illustrated in Fig. 4. It should be noted that customized LDP-ecGAN models can be employed for IDSs in industrial CPSs on demand. As shown in the model, there are three main modules, i.e., the generator, discriminator, and classifier, and each has an independent architecture as follows:

i) Generator $\mathscr{G}(\bullet)$: The generator hosts Fully-Connected (FC) layers, Dropout (Dt) layers, and LeakyReLU (LReLU) activation functions, and the FC layers are sorted from the smallest scale to the largest along with the last FC layer that contains the equal number of neurons as the dimension of input data. A randomly sampled vector $z \sim p(z)$ embedded with generated class information l (i.e., a certain class label) in the final dimension is inputted to condition $\mathscr{G}(\bullet)$ to generate the data sample of a certain class as $\hat{x} = \mathscr{G}(z|l)$ for semi-supervised learning.

ii) Wasserstein Discriminator $\mathscr{D} \triangleq f_\omega(\bullet)$: The discriminator uses a similar architecture to the generator, but lines up the FC combos from the largest scale to the smallest with no Dt layer, and it ends with a 1×1 dimension FC layer to differentiate synthetic samples from real data. $f_\omega(x|y) = 1$ and $f_\omega(\hat{x}|l) = 1$ indicate that the condition inputs, including the real condition input $(x,y) \sim p(S)$ and the synthetic one $(\hat{x}, l)$, are both classified as real by $f_\omega(\bullet)$. Oppositely, the condition input is considered as fake when $f_\omega(x|y) = 0$ or $f_\omega(\hat{x}|l) = 0$.

iii) Classifier $\mathscr{C}(\bullet)$: The classifier is mainly composed of three convolutional blocks, followed by two FC layers, a dropout layer, and a softmax layer. Each convolutional block consists of a 1-Dimensional Convolutional (Conv 1D) layer, a Batch Normalization (BN) layer, and a max-pooling layer. The softmax layer is exploited to map the nonnormalized output of $\mathscr{C}(\bullet)$ to a probability distribution over predicted classes (e.g., "Normal", "DoS", "Injection", etc.) in accordance with $y = \mathscr{C}(x)$ or $y = \mathscr{C}(\mathscr{G}(z|l))$.

The procedures of local training for LDP-ecGAN on the requester are summarized in Algorithm 1, and are visualized in Fig. 3(a) (**notice**: the analogical process of local training is applied on the providers, and has to be omitted here for the page limit). In the beginning, the requester initializes $\mathscr{G}$, $\mathscr{D}$, and $\mathscr{C}$ with weights ϕ, ω, and θ in the line 1. When $t_1 \leq T_\phi$, where T_ϕ is the iterations of $\mathscr{G}$, $\mathscr{G}$ and $\mathscr{D}$ are trained firstly in

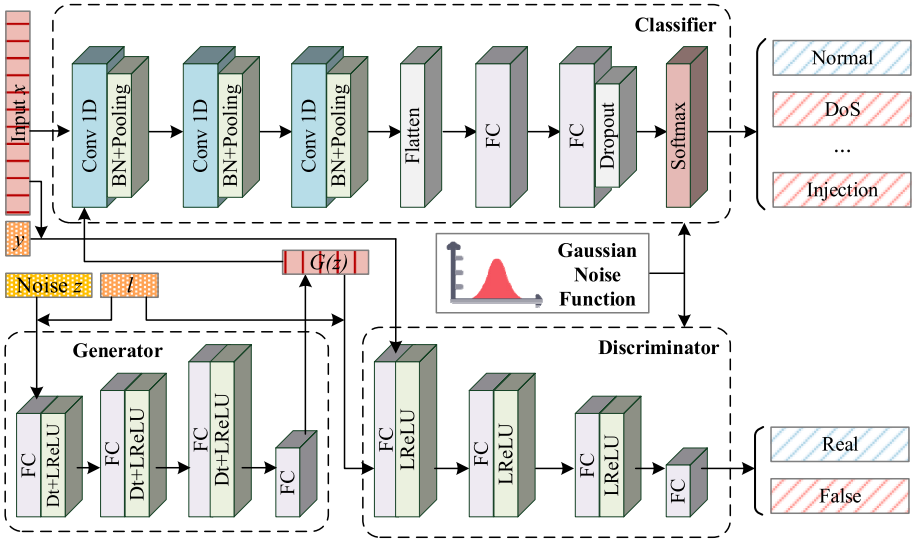

Fig. 4. Generic architecture of LDP-ecGAN model.

the lines 3–15 to generate practicable samples. In a loop (the lines 5–10), $\mathscr{D}$ is trained iteratively with $\{x^{(b)}, y^{(b)}\}_b^B$ and $\{z^{(b)}\}_b^B$, until $t_2 > T_\omega$. Specifically, when computing a $\mathscr{D}$'s gradient with respect to a real sample $(x^{(b)}, y^{(b)})$ and a random noise $z^{(b)}$, the gradient is pruned by injecting the well-designed Gaussian noise in the lines 7–8, ensuring that the sensitivity is bounded by ϵ. A $\mathscr{D}$'s gradient for each b can be obtained by Eq. (8):

$$g_\omega\left(x^{(b)}, y^{(b)}, z^{(b)}\right) = \nabla_\omega\left[f_\omega\left(x^{(b)}\middle|y^{(b)}\right) - f_\omega\left(\mathscr{G}\left(z^{(b)}\middle|y^{(b)}\right)\middle|y^{(b)}\right)\right] \tag{8}$$

where $f_\omega(\bullet)$ and $\mathscr{G}(\bullet)$ are conditioned on auxiliary information, i.e., a class label $y^{(b)}$. In the line 9, the RMSProp$(\bullet)$ (Wasserstein GAN, 2023) is an optimization function that can adaptively adjust ω_{t_2} (ω in t_2 epoch) according to the magnitude of the gradient $\widetilde{g}_\omega$. The clip$(\bullet)$ function (Xu et al., 2019) is further adopted to guarantee that $\{f_\omega\}_\omega$ are all K_ω-Lipschitz and act in a way to bound the gradient from each data in the line 10. Out of the loop, the average gradient of $\mathscr{G}$ is calculated with regard to $\left\{z^{(b)}, l^{(b)}\right\}_{b=1}^B$ in the line 14 as Eq. (9):

$$\overline{g}_\phi = -\nabla_\phi \frac{1}{B}\sum\nolimits_b f_\omega\left(\mathscr{G}\left(z^{(b)}\middle|l^{(b)}\right)\middle|l^{(b)}\right) \tag{9}$$

where $f_\omega(\bullet)$ and $\mathscr{G}(\bullet)$ are conditioned on a randomly generated label $l^{(b)}$, and then ϕ_{t_1} is updated to ϕ_{t_1+1} by using the RMSProp$(\bullet)$ with the learning rate α_ϕ and the average gradient $\overline{g}_\phi$ in the line 15. When $T_\phi < t_1 \le T_\theta$, the local training of the external classifier $\mathscr{C}$ is performed in the lines 16–24. In the line 20, every $\mathscr{C}$'s gradient $g_\theta\left(x^{(b)}, y^{(b)}, z^{(b)}, l^{(b)}\right)$ is computed for a pair of real samples $(x^{(b)}, y^{(b)})$ and synthetic data $\left(\mathscr{G}\left(z^{(b)}\middle|l^{(b)}\right), l^{(b)}\right)$ by minimizing the empirical loss function $\mathscr{L}(\theta)$ as Eq. (10):

$$g_\theta\left(x^{(b)}, y^{(b)}, z^{(b)}, l^{(b)}\right) = \nabla_\theta\Big[\mathscr{L}_{ce}\left(\mathscr{C}\left(x^{(b)}\right), y^{(b)}\right) + \lambda\mathscr{L}_{ce}\Big(\mathscr{C}\left(\mathscr{G}\left(z^{(b)}\middle|l^{(b)}\right)\right), \operatorname{argmax}\left(\mathscr{C}\left(\mathscr{G}\left(z^{(b)}\middle|l^{(b)}\right)\right)\right) > \tau\Big)\Big] \tag{10}$$

where $\mathscr{L}_{ce}(\bullet)$ is cross-entropy loss, τ is the pseudo-label threshold. Similarly, in the lines 21–23, we clip the norm of each g_θ along with the adding noise in order to protect the privacy and take a step in the opposite direction of this average noisy gradient $\widetilde{g}_\theta$ for updating θ_{t_1} to θ_{t_1+1}. The procedures above repeat iteratively until the desired level convergence for ϕ, ω, and θ is achieved. The computational complexity of Algorithm 1 primarily stems from its multi-layered loop structure and can be quantitatively expressed as $O\left(\max\left(T_\phi, T_\theta\right) \times T_\omega\right)$.

Algorithm 1 Local Training for LDP-ecGAN

Input: α_ϕ, α_ω, α_θ % Learning Rates of $\mathscr{G}$, $\mathscr{D}$ and $\mathscr{C}$
B % Batch Size of Data Sampling
T_ϕ, T_ω, T_θ % Iterations of $\mathscr{G}$, $\mathscr{D}$ and $\mathscr{C}$
σ_ω, σ_θ % Noise Scale of $\mathscr{D}$ and $\mathscr{C}$
C_ω, C_θ % Norm Bounds of $\mathscr{D}$ and $\mathscr{C}$
Output: Locally trained $\mathscr{G}$, $\mathscr{D}$ and $\mathscr{C}$ with DP
1: Initialize ϕ, ω and θ for $\mathscr{G}$, $\mathscr{D}$ and $\mathscr{C}$
2: **for** $t_1 = 1, 2, \ldots, T_\phi, \ldots, T_\theta$ **do**
3: **if** $t_1 \le T_\phi$ **then**
4: **for** $t_2 = 1, 2, \ldots, T_\omega$ **do**
5: Sample $\{z^{(b)}\}_{b=1}^B \sim p(z)$
6: Sample $\{x^{(b)}, y^{(b)}\}_{b=1}^B \sim p(S)$
7: for each b, compute $g_\omega\left(x^{(b)}, y^{(b)}, z^{(b)}\right)$ as Eq. (8)
8: $\widetilde{g}_\omega \leftarrow \frac{1}{B}\left(\sum_b g_\omega\left(x^{(b)}, y^{(b)}, z^{(b)}\right) + \mathscr{N}\left(0, \sigma_\omega^2 C_\omega^2 \mathbf{I}\right)\right)$
9: $\omega_{t_2+1} \leftarrow \omega_{t_2} + \alpha_\omega \bullet \text{RMSProp}(\omega_{t_2}, \widetilde{g}_\omega)$
10: $\omega_{t_2+1} \leftarrow \text{clip}(\omega_{t_2+1}, -C_\omega, C_\omega)$
11: **end for**

(*continued on next column*)

(*continued*)

Algorithm 1 Local Training for LDP-ecGAN

12: Sample $\{z^{(b)}\}_{b=1}^B \sim p(z)$
13: Generate $\left\{l^{(b)}\right\}_{b=1}^B$ randomly
14: Compute $\overline{g}_\phi$ according to Eq. (9)
15: $\phi_{t_1+1} \leftarrow \phi_{t_1} - \alpha_\phi \bullet \text{RMSProp}\left(\phi_{t_1}, \overline{g}_\phi\right)$
16: **else**
17: Sample $\{z^{(b)}\}_{b=1}^B \sim p(z)$
18: Sample $\{x^{(b)}, y^{(b)}\}_{b=1}^B \sim p(S)$
19: Generate $\left\{l^{(b)}\right\}_{b=1}^B$ randomly
20: for each b, compute $g_\theta\left(x^{(b)}, y^{(b)}, z^{(b)}, l^{(b)}\right)$ as Eq. (10)
21: $\widetilde{g}_\theta \leftarrow \frac{1}{B}\left(\sum_b g_\theta\left(x^{(b)}, y^{(b)}, z^{(b)}, l^{(b)}\right) + \mathscr{N}\left(0, \sigma_\theta^2 C_\theta^2 \mathbf{I}\right)\right)$
22: $\theta_{t_1+1} \leftarrow \theta_{t_1} - \alpha_\theta \widetilde{g}_\theta$
23: $\theta_{t_1+1} \leftarrow \text{clip}(\theta_{t_1+1}, -C_\theta, C_\theta)$
24: **end if**
25: **end for**

Our method focuses on building the LDP-ecGAN model with Wasserstein distance and label condition and enforcing LDP by injecting well-designed Gaussian noise in updating the discriminator and classifier. We aim to control the influence of the training data specifically in the stochastic gradient descent computation by adding the noise into each gradient with respect to the training data, computing the average of these gradients, and clipping the bound norm of the average noisy gradients. The weights of the discriminator and classifier can be shown to guarantee DP with respect to the training data, and the privacy of the data that have not been sampled for training is guaranteed naturally as replacing these data does not cause any change in output distribution (i.e., equivalent to the case $\epsilon = 0$). The weights of the generator can also guarantee DP with respect to the training data. This is because there is a post-processing property of DP (Xu et al., 2019), which proves that any mapping after a differentially private output does not invade privacy. It is safe for the generator to generate data after the local training since the mapping here is in fact the computation of weights of the generator and the output is the differentially private weights of the discriminator.

4.2. Global training via DFD collaboration scheme

The DFD collaboration scheme is designed, with which each CPS can build a comprehensive LDP-ecGAN model in a decentralized way by initiating the federated knowledge distillation (Shen et al., 2020)-(Hardy et al., 2019) as a cooperative service requester to learn the collective knowledge and insight of other LDP-ecGAN models from various collaboration-oriented CPSs or cooperative service providers without sharing an isomorphic neural network model. In the following, the DFD collaboration scheme (as shown in Fig. 3(b)) is detailly described in two global training phases, namely requester's $\mathscr{G}$ global training and requester's $\mathscr{D}$ & $\mathscr{C}$ global training.

Algorithm 2 Requester's $\mathscr{G}$ Global Training

1: **repeat:**
2: **procedure** Requester
4: **for** $n = 1, \ldots, N$ **do**
5: Sample $\{z^{(b)}\}_{b=1}^B \sim p(z)$
6: Generate $L_{n,t} = \left\{l^{(b)}\right\}_{b=1}^B$ randomly
7: Send $(\widehat{X}_{n,t} = \left\{\mathscr{G}\left(z^{(b)}\middle|l^{(b)}\right)\right\}_{b=1}^B, L_{n,t})$ to Provider n
8: **end for**
9: Receive $\left\{\left\{e_{n,t}^{(b)}\right\}_{b=1}^B\right\}_{n=1}^N$ from Provider n
10: **for** each $\phi_t^{(i)} \in \phi_t$ **do**

(*continued on next page*)

(*continued*)

Algorithm 2 Requester's $\mathscr{G}$ Global Training

11: $\Delta\phi_t^{(i)} \leftarrow \frac{1}{NB}\sum_{n=1}^{N}\sum_{\hat{x}^{(b)}\in\hat{X}_{n,t}} e_{n,t}^{(b)}\frac{\partial\hat{x}^{(b)}}{\partial\phi_t^{(i)}}$
12: $\phi_{t+1}^{(i)} \leftarrow \phi_t^{(i)} + \text{Adam}\left(\Delta\phi_t^{(i)}\right)$
13: **end for**
14: **end procedure**
15: **procedure** Provider n
16: Receive $(\hat{X}_{n,t}, L_{n,t}) = \left\{\hat{x}^{(b)}, l^{(b)}\right\}_{b=1}^{B}$ from Requester
17: **for** $b = 1, \ldots, B$ **do**
18: Compute $e_{n,t}^{(b)}$ according to Eq. (11)
19: **end for**
20: Send $\left\{e_{n,t}^{(b)}\right\}_{b=1}^{B}$ to Requester
21: **end procedure**
23: **until** ϕ_{t+1} is converged or $++t > T$

The pseudo-code of the global training for the requester's $\mathscr{G}$ is presented in Algorithm 2, whose computational complexity is calculated as $O(T \times (N+1))$, accounting for the iterations over T time steps and N cooperative service providers, plus the additional service requester. For every global iteration t, the requester's $\mathscr{G}$ firstly produces a batch of synthetic samples of size B, i.e., $\hat{X}_{n,t} = \left\{\mathscr{G}\left(z^{(b)}\middle|l^{(b)}\right)\right\}_{b=1}^{B}$, with randomly generated labels $L_{n,t} = \left\{l^{(b)}\right\}_{b=1}^{B}$. Then, each provider in CPSs is sent with $(\hat{X}_{n,t}, L_{n,t})$ for the computation of the gradients of $\mathscr{G}$. The error terms $\left\{e_{n,t}^{(b)}\right\}_{b=1}^{B}$ of the *n-th* provider can be calculated once receiving $(\hat{X}_{n,t}, L_{n,t})$ from the requester, as Eq. (11):

$$e_{n,t}^{(b)} = -\frac{\partial \log\left(f_{\omega_n}\left(\hat{x}^{(b)}\middle|l^{(b)}\right)\right)}{\partial\hat{x}^{(b)}} \tag{11}$$

where $\hat{x}^{(b)}$ is the *b-th* data of batch $\hat{X}_{n,t}$. When the requester obtains the error terms $\left\{\left\{e_{n,t}^{(b)}\right\}_{b=1}^{B}\right\}_{n=1}^{N}$ from all provides, the weight update of $\mathscr{G}$, i.e., $\Delta\phi_t = -\frac{\partial \log(f_{\omega n}(\hat{x}^{(b)}|l^{(b)}))}{\partial\phi_t}$, is deduced from all $\left\{\left\{e_{1,t}^{(b)}\right\}_{b=1}^{B}, \left\{e_{2,t}^{(b)}\right\}_{b=1}^{B}, \cdots, \left\{e_{N,t}^{(b)}\right\}_{b=1}^{B}\right\}$ as $\Delta\phi_t^{(i)} = \frac{1}{NB}\sum_{n=1}^{N}\sum_{\hat{x}^{(b)}\in\hat{X}_{n,t}} e_{n,t}^{(b)}\frac{\partial\hat{x}^{(b)}}{\partial\phi_t^{(i)}}$, where $\Delta\phi_t^{(i)}$ is the *i-th* element of $\Delta\phi_t$. After computing $\Delta\phi_t^{(i)}$, the Adam optimizer, the most common method to aggregate updates in parallel (Hardy et al., 2019)-(Ramadevi et al., 2022), is adopted to update $\phi_{t+1}^{(i)}$ as $\phi_{t+1}^{(i)} = \phi_t^{(i)} + \text{Adam}\left(\Delta\phi_t^{(i)}\right)$. The iterative process is recited continuously until ϕ_{t+1} is converged or the maximum number of iterations is reached. For the global training, the requester's $\mathscr{G}$ is updated using the providers' $\{\mathscr{D}_n\}_n$ and their local shares $\{S_n\}_n$. It is a 1-versus-N game, in which $\mathscr{G}$ is optimized to generate synthetic data considered as real by all the providers while every provider's $\mathscr{D}_n$ tries to differentiate the generated data of $\mathscr{G}$ from the real data in $\{S_n\}_n$.

For better understanding, the newly-defined notations related to Algorithm 3 are explained. $m \in M = \{$"*Normal*", "*DoS*", "*Injection*"$\ldots\}$ is assumed to be an alphabet of $|M|$ labels under consideration. The function $F\left(\theta_n, s_n^{(b)}\right)$ (or $F\left(\omega_n, s_n^{(b)}\right)$) is the logit vector normalized by the softmax function, where θ_n (or ω_n) and $s_n^{(b)} \in S_n$ are $\mathscr{C}_n$'s (or $\mathscr{D}_n$'s) weights and input respectively. The function $\mathscr{L}_{ce}(\bullet)$ is the cross-entropy loss that is used for both loss function and distillation regularizer. The term γ_θ (or γ_ω) is a weight parameter of θ (or ω). $\overline{F}_{\theta_n,t}^{m}$ (or $\{\overline{F}_{\omega_n,t}^{0}; \overline{F}_{\omega_n,t}^{1}\}$) is the local-average logit vector of θ_n (or ω_n) at the *t-th* iteration when the training sample belongs to the *m-th* (or $\{0,1\}$-*th*) ground truth label, $\overline{F}_t^m$ (or $\{\overline{F}_t^0, \overline{F}_t^1\}$) is the global-average logit vector that equals to the average of $\left\{\overline{F}_{\theta_n,t}^{m}\right\}_n$ (or $\left\{\left\{F_{\omega_n,t}^{0}\right\}_n, \left\{F_{\omega_n,t}^{1}\right\}_n\right\}$). $cnt_{\theta_n,t}^{m}$ (or $\left\{cnt_{\omega_n,t}^{0}, cnt_{\omega_n,t}^{1}\right\}$) counts the number of the samples whose ground-truth labels are m (or $\{0,1\}$). As shown in Algorithm 3, the global training on Requester's $\mathscr{D}$ & $\mathscr{C}$ at every *t-th* epoch mainly contains the following steps:

Step 1: the *n-th* provider utilizes its local dataset S_n to compute the local-aggregate logit vectors $\left\{F_{\theta_n,t}^{m}\right\}_m^M$ of $\mathscr{C}_n$ for every label m ($m \in M$), and counts the corresponding amount for each m into $cnt_{\theta_n,t}^{m}$, as Eqs. 12 and 13:

$$F_{\theta_n,t}^{m} = F_{\theta_n,t}^{m} + F\left(\theta_n, s_n^{(b)}\right) \tag{12}$$

$$cnt_{\theta_n,t}^{m} = cnt_{\theta_n,t}^{m} + 1 \tag{13}$$

Apart from S_n, a set of synthetic samples $\hat{S}_n = \left\{\hat{s}_n^{(b)}\middle|\hat{s}_n^{(b)} = \left\{\hat{x}_n^{(b)}, l_n^{(b)}\right\}\right\}_{b=1}^{B}$ is generated as a supplement to calculate the local-aggregate logit vectors $\left\{F_{\omega_n,t}^{0}, F_{\omega_n,t}^{1}\right\}$ of $\mathscr{D}_n$ and the counts $\left\{cnt_{\omega_n,t}^{0}, cnt_{\omega_n,t}^{1}\right\}$ for real and fake data by Eqs. 14 and 15:

$$F_{\omega_n,t}^{1} = F_{\omega_n,t}^{1} + F\left(\omega_n, s_n^{(b)}\right), F_{\omega_n,t}^{0} = F_{\omega_n,t}^{0} + F\left(\omega_n, \hat{s}_n^{(b)}\right) \tag{14}$$

$$cnt_{\omega_n,t}^{0} = cnt_{\omega_n,t}^{0} + 1, cnt_{\omega_n,t}^{1} = cnt_{\omega_n,t}^{1} + 1 \tag{15}$$

Step 2: the *n-th* provider uploads its all local-average logit vectors $\left(\left\{\overline{F}_{\omega_n,t}^{0}, \overline{F}_{\omega_n,t}^{1}\right\}, \left\{\overline{F}_{\theta_n,t}^{m}\right\}_m^M\right)$ calculated as Eqs. 16 and 17:

$$\overline{F}_{\theta_n,t}^{m} = \frac{F_{\theta_n,t}^{m}}{cnt_{\theta_n,t}^{m}} \tag{16}$$

$$\overline{F}_{\omega_n,t}^{0} = \frac{F_{\omega_n,t}^{0}}{cnt_{\omega_n,t}^{0}}, \overline{F}_{\omega_n,t}^{1} = \frac{F_{\omega_n,t}^{1}}{cnt_{\omega_n,t}^{1}} \tag{17}$$

Then, the requester averages the uploaded local-average logit vectors from all the providers separately for each label, and constructs the global-average logit vectors of all labels as $\overline{F}_t^0 = \frac{\sum_n \overline{F}_{\omega_n,t}^{0}}{N}, \overline{F}_t^1 = \frac{\sum_n \overline{F}_{\omega_n,t}^{1}}{N}$ and $\left\{\overline{F}_t^m = \frac{\sum_n \overline{F}_{\theta_n,t}^{m}}{N}\middle| m \in M\right\}$.

Algorithm 3 Requester's $\mathscr{D}$ & $\mathscr{C}$ Global Training

1: **repeat:**
2: **procedure** Provider n
3: **for** $s_n^{(b)} = \left(x_n^{(b)}, y_n^{(b)}\right) \in S_n$ **do**
4: $F_{\theta_n,t}^{m} \leftarrow F_{\theta_n,t}^{m} + F\left(\theta_n, s_n^{(b)}\right)$, $cnt_{\theta_n,t}^{m} += 1$
5: $F_{\omega_n,t}^{1} \leftarrow F_{\omega_n,t}^{1} + F\left(\omega_n, s_n^{(b)}\right)$, $cnt_{\omega_n,t}^{1} += 1$
6: **end for**
7: **for** $\hat{s}_n^{(b)} = \left(\hat{x}_n^{(b)}, l_n^{(b)}\right) \in \hat{S}_n$ **do**
8: $F_{\omega_n,t}^{0} \leftarrow F_{\omega_n,t}^{0} + F\left(\omega_n, \hat{s}_n^{(b)}\right)$, $cnt_{\omega_n,t}^{0} += 1$
9: **end for**
10: $\overline{F}_{\omega_n,t}^{0} \leftarrow \frac{F_{\omega_n,t}^{0}}{cnt_{\omega_n,t}^{0}}$, $\overline{F}_{\omega_n,t}^{1} \leftarrow \frac{F_{\omega_n,t}^{1}}{cnt_{\omega_n,t}^{1}}$
11: for each m, $\overline{F}_{\theta_n,t}^{m} \leftarrow \frac{F_{\theta_n,t}^{m}}{cnt_{\theta_n,t}^{m}}$
12: Send $(\left\{\overline{F}_{\omega_n,t}^{0}, \overline{F}_{\omega_n,t}^{1}\right\}; \left\{\overline{F}_{\theta_n,t}^{m}\right\}_m^M)$ to Requester
13: **end procedure**
14: **procedure** Requester
15: Receive $\left\{\left\{\overline{F}_{\omega_n,t}^{0}, \overline{F}_{\omega_n,t}^{1}\right\}, \left\{\overline{F}_{\theta_n,t}^{m}\right\}_m^M\right\}_{n=1}^{N}$
16: $\overline{F}_t^0 \leftarrow \frac{\sum_n \overline{F}_{\omega_n,t}^{0}}{N}$, $\overline{F}_t^1 \leftarrow \frac{\sum_n \overline{F}_{\omega_n,t}^{1}}{N}$

(*continued on next page*)

(*continued*)

Algorithm 3 Requester's $\mathscr{D}$ & $\mathscr{C}$ Global Training

17: for each m, $\overline{F}_t^m \leftarrow \frac{\sum_n \overline{F}_{\theta_n,t}^m}{N}$
18: for each b, compute $g_{\theta_t}(s^{(b)})$ by Eq. (18)
19: for each b, compute $g_{\omega_t}(s^{(b)}, \widehat{s}^{(b)})$ by Eq. (19)
20: for each g_{θ_t}, compute $\widehat{g}_{\theta_t}(s^{(b)})$ by Eq. (20)
21: for each g_{ω_t}, compute $\widehat{g}_{\omega_t}(s^{(b)}, \widehat{s}^{(b)})$ by Eq. (21)
22: $\widetilde{g}_{\theta_t} \leftarrow \frac{1}{B}\sum_b \left(\widehat{g}_{\theta_t}(s^{(b)}) + \mathscr{N}(0, \sigma_\theta^2 C_\theta^2 \mathbf{I})\right)$
23: $\widetilde{g}_{\omega_t} \leftarrow \frac{1}{B}\sum_b \left(\widehat{g}_{\omega_t}(s^{(b)}, \widehat{s}^{(b)}) + \mathscr{N}(0, \sigma_\omega^2 C_\omega^2 \mathbf{I})\right)$
24: $\omega_{t+1} \leftarrow \omega_t - \alpha_\omega \widetilde{g}_{\omega_t}$, $\theta_{t+1} \leftarrow \theta_t - \alpha_\theta \widetilde{g}_{\theta_t}$
25: **end procedure**
26: **until** $\omega_{t+1}, \theta_{t+1}$ are converged or $++t > T$

Step 3: As similar to Session III.A, the LDP technology is leveraged in updating the requester's $\mathscr{D}$ & $\mathscr{C}$. The gradient g_{θ_t} of $\mathscr{C}$ per $s^{(b)} \in S$ and the gradient g_{ω_t} of $\mathscr{D}$ for each $(s^{(b)}, \widehat{s}^{(b)})$ are computed with using $\left(\overline{F}_t^0, \overline{F}_t^1, \{\overline{F}_t^m\}_m^M\right)$, as Eqs. 18 and 19:

$$g_{\theta_t}(s^{(b)}) = \nabla_\theta \left[\mathscr{L}_{ce}\left(F(\theta_t, s^{(b)}), m\right) + \gamma_\theta \mathscr{L}_{ce}\left(F(\theta_t, s^{(b)}), \overline{F}_t^m\right)\right] \tag{18}$$

$$g_{\omega_t}(s^{(b)}, \widehat{s}^{(b)}) = \nabla_\omega \begin{bmatrix} \mathscr{L}_{ce}\left(F(\omega_t, s^{(b)}), 1\right) + \mathscr{L}_{ce}\left(F(\omega_t, \widehat{s}^{(b)}), 0\right) \\ +\gamma_\omega \left(\mathscr{L}_{ce}\left(F(\omega_t, s^{(b)}), \overline{F}_t^1\right) + \mathscr{L}_{ce}\left(F(\omega_t, \widehat{s}^{(b)}), \overline{F}_t^0\right)\right) \end{bmatrix} \tag{19}$$

Since there is no a priori bound on the size of the gradients, each g_{θ_t} and g_{ω_t} in L2-norm are clipped to $\widehat{g}_{\theta_t}$ and $\widehat{g}_{\omega_t}$ with their clipping thresholds C_θ and C_ω as Eqs. 20 and 21:

$$\widehat{g}_{\theta_t}(s^{(b)}) = \frac{g_{\theta_t}(s^{(b)})}{\max\left(1, \frac{\|g_{\theta_t}(s^{(b)})\|_2}{C_\theta}\right)} \tag{20}$$

$$\widehat{g}_{\omega_t}(s^{(b)}, \widehat{s}^{(b)}) = \frac{g_{\omega_t}(s^{(b)}, \widehat{s}^{(b)})}{\max\left(1, \frac{\|g_{\omega_t}(s^{(b)}, \widehat{s}^{(b)})\|_2}{C_\omega}\right)} \tag{21}$$

Step 4: In order to protect privacy, the requester adds the well-designed Gaussian noise, i.e., $\mathscr{N}(0, \sigma_\theta^2 C_\theta^2 \mathbf{I})$ and $\mathscr{N}(0, \sigma_\omega^2 C_\omega^2 \mathbf{I})$, to each g_{θ_t} and g_{ω_t} when computing the average noisy gradients $\widetilde{g}_{\theta_t}$ and $\widetilde{g}_{\omega_t}$. At last, the weights ω_t and θ_t of $\mathscr{D}$ & $\mathscr{C}$ are updated in the opposite direction of $\widetilde{g}_{\theta_t}$ and $\widetilde{g}_{\omega_t}$ as $\omega_{t+1} = \omega_t - \alpha_\omega \widetilde{g}_{\omega_t}$ and $\theta_{t+1} = \theta_t - \alpha_\theta \widetilde{g}_{\theta_t}$. Similar to Algorithm 2, the computational complexity of this process is also quantified as $O(T \times (N+1))$, which reflects the iterations over T time steps and operations involving $N+1$ cooperative participants.

The DFD collaboration scheme can enhance the classification performance of the requester's discriminator and classifier model since the predictions of the providers' models are able to give more helpful information (soft targets) than one-hot labels (hard targets) as a regularizer (Shen et al., 2020), which will be proved in Section 6 and the theory analysis will be discussed in detail.

5. QoS-HLF framework

5.1. Problem statement

In light of the quest for enhanced performance, computational efficiency, and addressing data heterogeneity, selectively aggregating models based on specific criteria or performance indicators might be a more ideal strategy. To empirically validate this assumption, a detailed comparative analysis between FL-based and centralized Stochastic Gradient Descent (SGD) models is undertaken, which focuses on two distinct aggregation methods: random model aggregation and performance-weighted aggregation. Since the performance is determined by the model weights obtained through training, comparing FL-based with centralized SGD models involves quantifying the weight differences between different models under the same initial conditions.

Taking the classifier weights of LDP-ecGAN, i.e., θ, as an example. To determine θ, centralized SGD addresses the optimization challenge through iterative updates. Let's denote $\theta_{c,t}$ as the weights following the t-th update within a centralized framework. Accordingly, the update process executed by centralized SGD can be elucidated by Eq. (22):

$$\theta_{c,t} = \theta_{c,t-1} - \eta \nabla_{\theta_c} \mathscr{L}_{ce}(\theta_{c,t-1}) = \theta_{c,t-1} - \eta \sum\nolimits_x p_{\theta_c} \nabla_{\theta_c} \mathbb{E}_x \left[\log f_{\theta_c}(x, \theta_{c,t-1})\right] \tag{22}$$

Adhere to the previous assumption, there are N cooperative service providers $\{v_n\}_n^N$, with the n-th provider equipped with a local dataset S_n, where $|S_n|$ represents the size of its dataset. In the federated learning, let's consider that each cooperative service provider undertakes local SGD operations autonomously on their respective datasets. During the t-th iteration over S_n, the operation executed by local SGD is illustrated as shown in Eq. (23):

$$\theta_{n,t} = \theta_{n,t-1} - \eta \sum\nolimits_x p_{\theta_n} \nabla_{\theta_n} \mathbb{E}_x \left[\log f_{\theta_n}(x, \theta_{n,t-1})\right] \tag{23}$$

When synchronization occurs after every T steps, the weights of FL-based model, i.e., LDP-ecGAN with DFD collaboration, after the m-th synchronization, i.e., $\theta_{f,mT}$, can be derived as Eq. (24):

$$\theta_{f,mT} = \sum\nolimits_{n=1}^N \frac{|S_n|}{\sum_{n=1}^N |S_n|} \theta_{n,mT} \tag{24}$$

An example of the variation in the difference between $\theta_{c,mT}$ and $\theta_{f,mT}$ is illustrated in Fig. 5. It's noteworthy that the performance-weighted aggregation can significantly mitigate the difference between $\theta_{c,mT}$ and $\theta_{f,mT}$ by excluding the providers v_1 and v_2. This approach emphasizes the

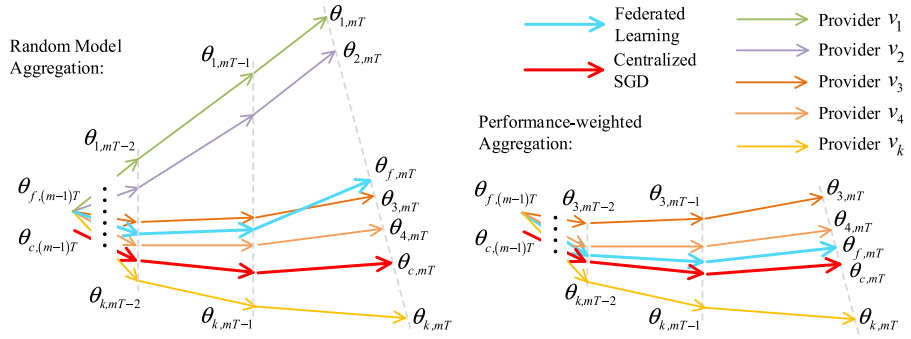

Fig. 5. Weight differences between FL-based and centralized SGD models.

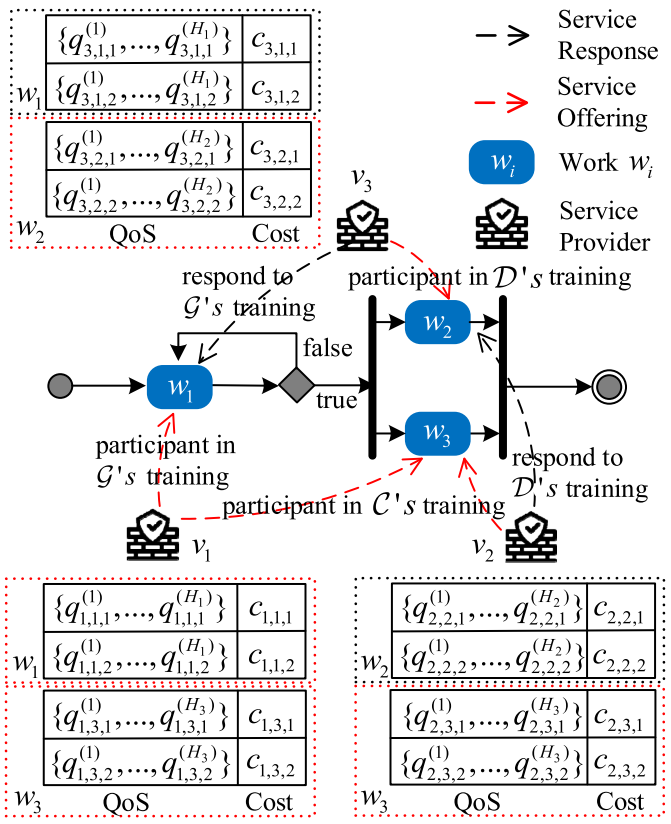

Fig. 6. QoS-aware collaboration of PB-fdGAN.

integration of model updates based on their individual performance, ensuring that more accurate and reliable models have a greater influence on the final aggregated model. It prevents the dilution of model performance that can result from uniformly aggregating disparate updates, some of which may stem from non-representative data or even adversarial sources. Therefore, the QoS-HLF framework is crafted, which incorporates an innovative QoS evaluation methodology to select preferred intrusion detection models for performance-weighted aggregation and constructs customized CCs to guarantee the reliable execution of DFD collaboration.

5.2. QoS evaluation methodology

According to Section 4.2, the collaboration with the assistance of multiple CPSs can be modeled as a QoS-aware service composition problem, called the QoS-aware collaboration of PB-fdGAN. Suppose that an industrial CPS or the cooperative service requester decided to carry out the global training to improve its IDS model through the use of local datasets collected by third-party CPSs, i.e., the cooperative service providers. The QoS-aware collaboration is illustrated in Fig. 6. The requester decomposes the global training process into three works w_1 (*requester's* $\mathcal{G}$ *global training*), w_2 (*requester's* $\mathcal{D}$ *global training*) and w_3 (*requester's* $\mathcal{C}$ *global training*). First, the work w_1 is executed to collectively train $\mathcal{G}$; if $\mathcal{G}$ is not well-trained for semi-supervised learning, w_1 is executed again; otherwise, w_2 and w_3 are performed simultaneously to strengthen the classification performance of $\mathcal{D}$ & $\mathcal{C}$ along with using the synthetic samples of $\mathcal{G}$. Assume there are K candidate service providers $V = \{v_k\}_{k=1}^{K}, K \leq N$. Each v_k could perform a work w_i with various plans $\left\{p_j\right\}_j$ and have flexible QoSs, i.e., $q_{k,i} = \left\{q_{k,i,j}\right\}_j$, and $q_{k,i,j} = \left\{q_{k,i,j}^{(h)}\right\}_h^{H_i}$ where h represents the *h-th* QoS attribute. An individual QoS with a different plan towards a work requires a different cost, e.g., the cost of $q_{k,i,j}$ is $c_{k,i,j}$. In practice, the service providers use dynamic pricing strategies. The requester and providers require negotiation to reach an agreement on QoSs and prices. For example, for a work w_i, the service requester u proposes that the desire QoS is $q_{u,i} = \left\{q_{u,i}^{(h)}\right\}_h^{H_i}$ and the maximum price is $c_{u,i}$. Qualified providers propose that their QoSs are $\left\{q_{k,i,j} = \left\{q_{k,i,j}^{(h)}\right\}_h^{H_i}\right\}_{k,j}$, and the tender prices are $\left\{c_{k,i,j}\right\}_{k,j}$. They can reach an agreement for the work w_i as $a_i = \left\{\left(u, v_k, w_i, q_{k,i,j}, c_{k,i,j}\right) \middle| q_{k,i,j} \geqslant q_{u,i}, \sum_k \sum_j c_{k,i,j} \leq c_{u,i}\right\}$, and the actual cost is c_i ($c_i = \sum_k \sum_j c_{k,i,j}$). A QoS $q_{k,i,j}$ satisfies the QoS constraints $q_{u,i}$ (denoted by $q_{k,i,j} \geqslant q_{u,i}$), iif: $\forall h \in [1, H_i]$.

$$\begin{cases} q_{k,i,j} \geq q_{u,i}, z_h = 1 \\ q_{k,i,j} \leq q_{u,i}, z_h = 0 \end{cases} \tag{25}$$

where $z_h = 1$ if the *h-th* QoS attribute is a positive attribute, and $z_h = 0$ if the *h-th* QoS attribute is a negative attribute.

Normally, a service provider usually has many QoS attributes. As for evaluating a service provider based on multiple QoS attributes, its QoS must be transformed into a single value using the simple additive weighting technique. The QoS of a service provider v_k adopting a plan p_j for a work w_i is $q_{k,i,j} = \left\{q_{k,i,j}^{(1)}, q_{k,i,j}^{(2)}, \ldots, q_{k,i,j}^{(H_i)}\right\}$ that can be normalized as:

$$\eta\left(q_{k,i,j}^{(h)}\right) = \begin{cases} \dfrac{q_{k,i,j}^{(h)} - q_{i,min}^{(h)}}{q_{i,max}^{(h)} - q_{i,min}^{(h)}}, z_h = 1 \wedge q_{i,max}^{(h)} \neq q_{i,min}^{(h)} \\ \dfrac{q_{i,max}^{(h)} - q_{k,i,j}^{(h)}}{q_{i,max}^{(h)} - q_{i,min}^{(h)}}, z_h = 0 \wedge q_{i,max}^{(h)} \neq q_{i,min}^{(h)} \\ 1, q_{i,max}^{(h)} = q_{i,min}^{(h)} \end{cases} \tag{26}$$

where $q_{i,min}^{(h)}$ and $q_{i,max}^{(h)}$ are the minimum and maximum values of the *h-th* QoS attribute for the work w_i. The QoS $q_{k,i,j}$ can be transformed into a single value for comparison as below:

$$\eta\left(q_{k,i,j}\right) = \sum_{h=1}^{H_i} \left(\mu_h \times \eta\left(q_{k,i,j}^{(h)}\right)\right) \tag{27}$$

$$o\left(q_{k,i,j}\right) = b \times \left(\eta\left(q_{k,i,j}\right)\right)\hat{d} \tag{28}$$

where μ_h is the parameter that represents the service requester's priority for the *h-th* QoS attribute ($\sum_{h=1}^{H_i} \mu_h = 1$), b and d are the coefficients. A higher QoS brings more benefits to the requester. However, a higher QoS may require a higher price. For any work in the QoS-aware collaboration, the utility of the requester is considered as the difference between the service QoS valuation and the amount of money it spends. The utility function of the requester to reach an agreement a_i on a work w_i with the widest possible diversity of the service providers can be defined as follows:

$$\mathcal{U}(a_i) = \sum\nolimits_k \max_j \left(o\left(q_{k,i,j}\right) - c_{k,i,j}\right) \tag{29}$$

Such that, the objective of the service requester is to reach an agreement for all works to achieve an optimization goal (e.g., utility maximization) as follows:

$$\begin{cases} \textit{Maximize } \mathcal{U}(A) = \sum\limits_i \mathcal{U}(a_i) \\ s.t. \quad \forall\, a_i \neq \varnothing \textit{and} \sum\limits_i c_i \leq \textit{budget} \end{cases} \tag{30}$$

Actually, the cooperative service requester and providers are autonomous and self-interested. Even if they have reached an agreement, they may gain more utility than usual from falsification. Two critical problems have to be addressed to achieve the utility

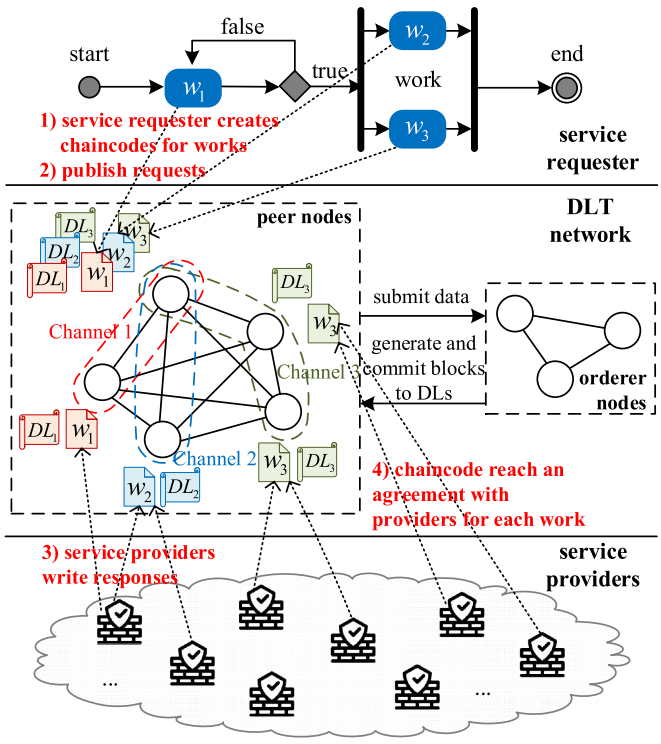

Fig. 7. Proposed QoS-HLF framework.

maximization; **i)** *when the service providers change dynamically, how does the requester find and reach an agreement with other cost-efficient substitutes at runtime?* **ii)** *when there is no trusted central organization, how do the service requesters and providers implement their agreements?* Thus, in this paper, the QoS-HLF Framework is crafted to solve the two mentioned problems.

5.3. QoS-HLF framework

Our proposed QoS-HLF framework is designed based on a permissioned Distributed Ledger Technology (DLT), i.e., HyperLedger Fabric or HLF. It is a platform for distributed ledger solutions, underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility, and scalability (Hyperledger Fabric, 2023). The reasons why HLF is adopted in the proposed framework are because i) there is no Power of Work (PoW) and crypto mining algorithm in HLF, hence the transaction throughput is several times those of public blockchains (e.g., Bitcoin and Ethereum) (Mothukuri et al., 2021); ii) the CCs of HLF, which handle business logic agreed to by HLF members, run with the properties of multithreaded communication and synchronization, so that a CC can be invoked to update or query the Distributed Ledger (DL) of HLF in a proposal transaction even if other CCs have not completed; iii) the execution results of CCs are still voted on before they are added into the DLs.

The proposed QoS-HLF framework is shown in Fig. 7. In the framework, there are two types of nodes. *peer*: the nodes are connected with each other creating a DLT network, which utilizes CCs to initialize and manage the DL states through the transactions submitted by applications; *orderer*: the nodes that provide the consensus service. The peer nodes cannot generate blocks. They need to submit the data to the orderer nodes, and then the orderer nodes generate a sequence of ordered blocks and deliver the blocks to all peer nodes. Each peer node independently adds the blocks into the respective DL, but in exactly the same way as every other peer node. By this means, all the DLs can be kept consistent. A CC is a set of program codes that run on the peer nodes of the DLT network and its execution results can be added into the corresponding DL. For providing an efficient sharing of infrastructure while maintaining data and communication privacy, channels are created among peer nodes, and the service requester can separate the work traffic with different service providers into different channels. An individual channel with its own CCs and DL is a private "subnet" of communication between two or more specific nodes, and the dissemination of data, which includes the information of transactions, ledger state, and channel membership, is restricted to the permissioned peers in the channel.

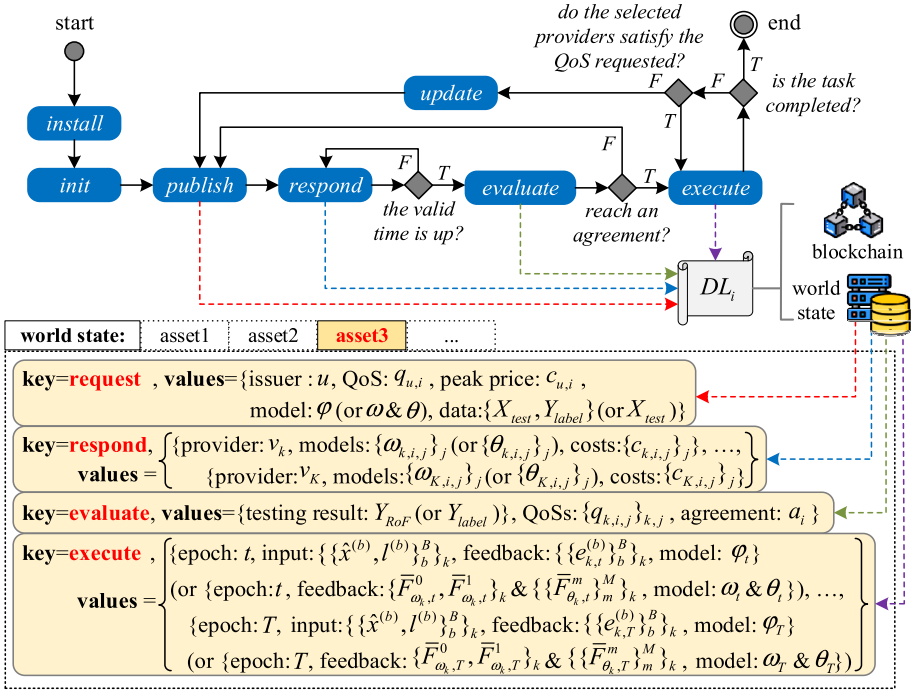

Fig. 8. QoS-Aware collaboration in HLF-QoS framework.

Fig. 7 also presents an example of the QoS-aware collaboration on the QoS-HLF framework. A collaboration-oriented CPS or the service requester creates and deploys an individual CC for each decomposed work within a separate channel on a peer node. For each work, the requester invokes the corresponding CC to publish its requests (e.g., the QoS, service price, etc.), and the copies of the CC are propagated within the channel and run on the nodes the service providers can interact with. The service providers read the requests, and subsequently write their responses into the copies of the CC. If there are one or more providers whose responses can satisfy the request, the CC automatically chooses the providers with superior QoSs and lower prices according to the predefined selection & pricing objective as Eq. (30), and creates an agreement with the selected providers for executing the global training process.

In the QoS-HLF framework, the service requester creates and utilizes an individual CC for each decomposed work of the global training process. Every CC has the functions: init, publish, respond, evaluate, execute, and update, but the functions with the same name differ in the implementation for different works. For a work w_i, an instance of the lifecycle of a CC and the CC recording the critical information into an asset of the DL are shown in Fig. 8, as follows:

i) init(): For a work w_i, the service requester u uses the install command to deploy an elaborate CC on the QoS-HLF framework and invokes the init function to initialize the CC.

ii) publish$\Big(u,\ q_{u,i},\ c_{u,i},\ \{X_{test},\ Y_{label}\}\ (or\ X_{test}),\ \phi\ (or\ \omega\ \&\ \theta)\Big)$: The requester u invokes the publish function to propose its requests, mainly including the expected QoS $q_{u,i}$, the maximum price $c_{u,i}$, the testing samples $\{X_{test}, Y_{label}\}$ for ϕ and ω (or X_{test} for θ), and the initial model ϕ (*or* ω & θ), into a newly-created asset for registering the critical information of w_i, and sets a valid time for the requests. Specifically, X_{test} is combined with real data and synthetic samples, Y_{label} is the labels of X_{test}, and $q_{u,i}$ is composed of the recognition criteria, such as *Precision*, *Recall*, *F1-Score*, and *False Alarm Rate*.

iii) respond$\Big(\{c_{k,i,j}\}_j, \{\omega_{k,j}\}_j\ \big(\text{or}\ \{\theta_{k,j}\}_j\big)\Big)$: In the valid time, a candidate service provider v_k invokes the respond function with providing multifarious local models $\{\omega_{k,j}\}_j$ (or $\{\theta_{k,j}\}_j$) and costs $\{c_{k,i,j}\}_j$ for different plans $\{p_j\}_j$. The response results will be recorded in the asset for service provider selection.

iv) evaluate(Y_{RoF} (or Y_{label})): If the valid time is up, the CC automatically checks whether there are satisfactory responses from the service providers. If yes, the evaluate function is invoked. For every v_k, given the testing results Y_{RoF} for $\{X_{test}, Y_{label}\}$ (or Y_{label} for X_{test}), the QoSs $\{q_{k,i,j}\}_j$ can be simply obtained by using $< \{X_{test}, Y_{label}\}, Y_{RoF} >$ (or $<X_{test}; Y_{label} >$) to test v_k's local model. An agreement a_i with maximum $\mathscr{U}(a_i)$ is attained after computing and evaluating the measurements for provider competition based on Eqs. (25)–(29), so as to include as much the service providers holding superior QoSs and competitive prices as possible. After that, the testing results, computed QoSs, and agreement are listed into the asset for reference.

v) execute$\Big(\Big\{\Big\{\hat{x}^{(b)}, l^{(b)}\Big\}_b^B\Big\}_k,\ \Big\{\Big\{e_{k,t}^{(b)}\Big\}_b^B\Big\}_k,\ \phi_t\ \Big(\text{or}\Big\{\Big\{\overline{F}_{\theta_k,t}^m\Big\}_m^M\Big\}_k,\ \theta_t\ \&\ \Big\{\overline{F}_{\omega_k,t}^0, \overline{F}_{\omega_k,t}^1\Big\}_k, \omega_t\Big)\Big)$: The requester invokes the execute function to implement the agreement with the selected providers. To be specific, for w_2 (or w_3), the execute function acquires the local-average logit vectors $\Big\{\Big\{\overline{F}_{\theta_k,t}^m\Big\}_m^M\Big\}_k$ (or $\Big\{\overline{F}_{\omega_k,t}^0, \overline{F}_{\omega_k,t}^1\Big\}_k$) from all selected providers at epoch t. When receiving enough feedbacks, the execute function forwards the received $\Big\{\Big\{\overline{F}_{\theta_k,t}^m\Big\}_m^M\Big\}_k$ (or $\Big\{\overline{F}_{\omega_k,t}^0, \overline{F}_{\omega_k,t}^1\Big\}_k$) to the service requester to update θ_t (or ω_t) and logs them into the asset for reference. For w_1, after distributing the generated $\Big\{\Big\{\hat{x}^{(b)}, l^{(b)}\Big\}_b^B\Big\}_k$ to all selected providers, the service requester's ϕ_t at epoch t is updated based on the received $\Big\{\Big\{e_{k,t}^{(b)}\Big\}_b^B\Big\}_k$. Simultaneously the related parameters are sent to the execute function and logged into the asset for audit. Once ϕ_t (or θ_t & ω_t) is converged or the maximum epoch is reached, the CC automatically pays each selected provider v_k the promised money $c_{k,i,j}$.

vi) update(): During the runtime of the execute function, if the CC detects that any selected provider is no longer available or failed to keep its QoS, the update function is automatically invoked enabling the service requester and qualified providers to propose a new request and responses and further reach a new agreement.

Table 2
Summary of simulation parameters.

Parameters	Values
Method of Cryptography	EC-DSA
Hash Function	SHA256
Learning Rate, $\alpha_{\phi/\omega/\theta}$	5×10^{-4}
Prune Constant, C_ω, C_θ	0.01
Local Iterations, $T_{\lfloor\phi,\omega,\theta\rfloor}$	[50, 5, 100]
Global Iterations, T	$T = T_\phi = T_\theta/2$
Latent Dimension, *dim* (z)	24
Batch Size of Sampling, B	$B = \lvert S\rvert/T$
Unsupervised Loss-Weight, λ	0.1
Pseudo-Label Threshold, τ	0.7
DP Noise Scale, (ϵ, δ)	(6, 1×10^{-5})

For the QoS-HLF framework, the requests, responses, and digital signatures from the service requesters and providers are submitted to the orderer nodes that provide a consensus service. The digital signatures ensure that the service requesters and providers cannot deny the requests and responses they proposed while any service requester or provider cannot falsify other peers' requests and responses. The consensus service ensures the consistency of the requests and responses on all peer nodes. Moreover, any agreements the service requester and multiple providers reach are implemented via the corresponding CCs, which can determine the real QoSs of the service providers to prevent dishonest ones from exaggerating the performance of their models, log all critical service information for audit, and automatically transfer the promised money to the service providers. Further, the sensitive information for a work/service, i.e., testing samples and results, model weights, and service prices, are only reachable for the authenticated and authorized participants in an isolated channel, and the LDP is enforced to all public models as discussed in Section 4. Therefore, the proposed QoS-HLF framework can not only assure the security and privacy of collaboration but also is resistant to poisoning and inference attacks for IDSs in industrial CPSs.

6. Performance evaluation

In this section, a comprehensive evaluation on the performance of our proposed PB-fdGAN is presented. The effectiveness of LDP-ecGAN mode, the DFD collaboration scheme, and QoS-HLF framework are firstly discussed, followed by the comparison of PB-fdGAN against the other state-of-the-art IDS solutions. For the page limit, the security analysis about the embedded LDP and QoS-HLF framework has to be shortly discussed in the last paragraphs of Sections 4.1, 5.3, and 6.4.

6.1. Simulation environment

Our experiments are conducted in a local Fabric network with the structure as presented in Fig. 7, including 3 orderer nodes and 5 peer nodes in an organization. Every two or three of them join a single channel. Each node is implemented as a Docker container in an

Table 3
WSTS results in effectiveness.

IDS Solutions	*Precision*	*Recall*	*F1-Score*	*FAR*
i) Without Local Differential Privacy				
CNN	95.29	94.69	94.88	5.33
ecGAN	96.49	95.46	95.80	3.26
DFD Collab.	97.21	96.89	96.93	2.82
PB-fdGAN	99.00	98.99	98.97	0.79
ii) With Local Differential Privacy				
CNN	93.25	93.33	93.14	7.90
LDP-ecGAN	95.68	94.10	94.67	4.62
DFD Collab.	96.36	95.68	95.87	3.32
PB-fdGAN	98.16	97.97	97.99	1.47

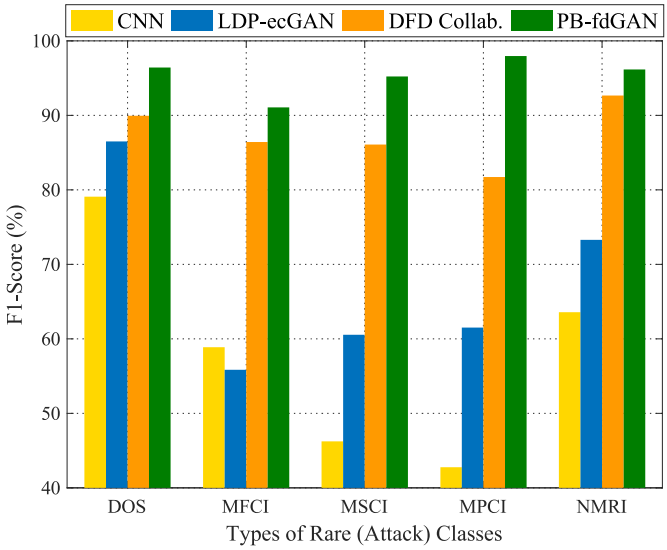

Fig. 9. *F1-Score* of different types of rare classes with LDP under WSTS.

independent physical server with a 2.60 GHz Intel(R) Xeon(R) CPU E5-2650 v2 and 8 GB of RAM, and each server runs Ubuntu 16.04 TLS with Hyperledger Fabric 2.0 and Tensorflow-CPU 2.7.0 installed. All physical servers are connected to the Fabric network using Docker Swarm through a 2000 Mbps Ethernet switch. The CCs are written in the language Go, and the language Node.js is used for CCs creation, deployment, and invocation. The scenario parameters are summarized in Table 2, in which the key experimental parameters will be discussed further in Section 6.4.

In the experiment, a real data resource, i.e., Water Storage Tank System (WSTS, one significant example of industrial CPSs) dataset (Industrial Control System (ICS), 2023) is adopted, which acts as not only a reliable tested dataset for intrusion detection experiments in industrial CPSs but also valuable information about the condition that takes place on a singling level when the different types of attacks occur on a real cyber-physical system. Each packet in the WSTS dataset collection is comprised of a vector of 23 attributes with the last attribute being the corresponding class or label. Every label determines the packet without any attack ("Benign" (73%)) or one type of attacks, i.e., "CMRI (5.5%)", "DOS (0.5%)", "MFCI (0.6%)", "MSCI (0.7%)", "MPCI (1.5%)", "NMRI (3.6%)", and "Reconnaissance (14.6%)". Notably, the classes, i.e., DOS, MFCI, MSCI, MPCI, and NMRI, comprise less than 4% of the WSTS dataset each, and thus categorized as "rare classes". This categorization underscores a considerable class imbalance, presenting unique challenges for IDSs in industrial CPSs. It necessitates a more targeted analysis of these rare classes to effectively address these challenges.

To extend the analysis and enhance the detection of a wider array of network intrusion attacks, the Aegean WiFi Intrusion Dataset (AWID) (Chatzoglou et al., 2021) is also employed to further affirm the efficacy

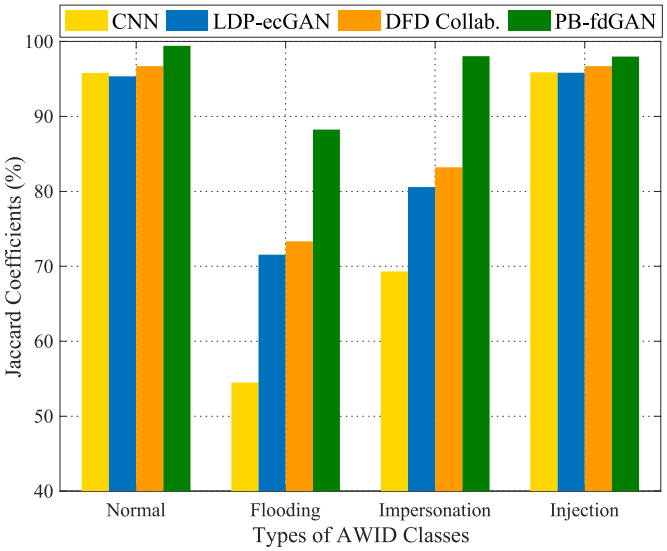

Fig. 10. *F1-Score* of different Jaccard coefficients with LDP under AWID.

of our proposed PB-fdGAN. AWID's category distribution consists of Normal (91%), Flooding (3.6%), Impersonation (2.7%), and Injection (2.7%). These figures indicate a similar sparsity in attack types as seen in WSTS, with attack categories constituting less than 4% of the total dataset. In the simulation experiments, both the WSTS and AWID datasets are segmented into 80% for training and 20% for testing. The training datasets are distributed equally among five different peer nodes within an industrial CPS, fostering a collaborative training environment for the intrusion detection system. The evaluation of the models is uniformly carried out using the same test datasets, with the effectiveness of the trained models being assessed by averaging their performance metrics across all scenarios, thereby ensuring a comprehensive evaluation of the proposed PB-fdGAN.

6.2. *Effectiveness evaluation*

In order to demonstrate the effectiveness of our PB-fdGAN, we monitor the performance among the Convolutional Neural Networks (CNN) with no proposed enhancement, the LDP-ecGAN model with progressively enhanced generative capabilities, the DFD collaboration scheme with an arbitrarily-aggregated model, and the proposed PB-fdGAN with our composite contributions. The results are presented under two different scenarios, i.e., with or without LDP technology, and the comparison is carried out in terms of *Precision*, *Recall*, *F1-Score*, and *False Alarm Rate (FAR)* in Table 3. In both situations (i.e., *w* or *w/o* LDP), the *Precision*, *Recall*, *F1-Score*, and *FAR* of PB-fdGAN are the most advantageous, and those of the DFD collaboration and LDP-ecGAN reach significant increments than those of CNN. Furthermore, Fig. 9 shows the comparison among the different IDS solutions for the rare classes, in which *F1-Score* is adopted since the metric is the harmonic mean and can symmetrically represent both *Precision* and *Recall* in one measurement. It is obvious that the *F1-Score* for the rare classes, i.e., DOS, MFCI, MSCI, MPCI, and NMRI, gets evident improvement when adopting the contributions in this paper, and the proposed PB-fdGAN achieves the highest values of *F1-Score* and the best performance on the minority classes.

Additionally, the AWID dataset is utilized to conduct a comparative experiment among CNN, LDP-ecGAN, DFD Collaboration, and PB-fdGAN. This evaluation is specifically designed to analyze and validate the effectiveness of these IDS solutions against a diverse spectrum of network intrusion attacks. The Jaccard coefficients ($\{J_i\}_i$, where J_i is calculated for each attack category) serve as a metric to assess the classification performance of IDS solutions in detecting various types of attacks. For each category within the AWID dataset, its Jaccard coeffi-

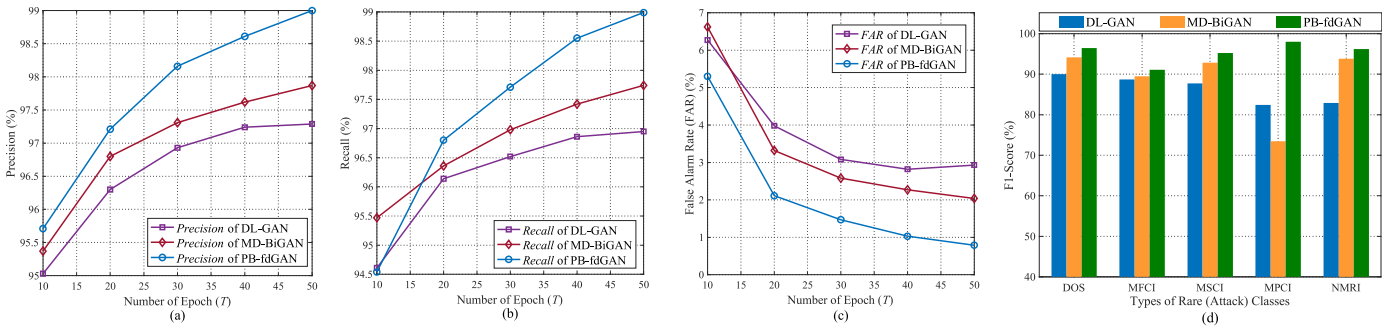

Fig. 11. Comparison among three IDS solutions under WSTS: (a) *Precision*, (b) *Recall*, (c) *FAR*, and (d) *F1-Score* among three IDS solutions.

Table 4
Comparison among three IDS solutions under AWID.

IDS Solutions	DL-GAN	MD-BiGAN	PB-fdGAN
J_{normal} (%)	96.80	98.11	99.41
$J_{flooding}$ (%)	54.28	74.35	88.23
$J_{impersonation}$ (%)	96.39	82.97	98.04
$J_{injection}$ (%)	95.65	96.62	97.98

cient relative to the entire dataset is calculated to measure the accuracy and robustness of IDS solutions to the corresponding attack category. A higher Jaccard coefficient for a specific attack type signifies that the IDS solution is particularly effective at identifying that kind of threat. The comparative results, illustrated in Fig. 10, show a clear progressive improvement from CNN through to PB-fdGAN, with PB-fdGAN achieving the highest performance metrics across all categories. Notably, PB-fdGAN demonstrates significant enhancements in detecting sparse attack categories, i.e., $J_{flooding}$, $J_{injection}$, and $J_{impersonation}$, underscoring its advanced capability in identifying these challenging types of threats.

The aforementioned results can be attributed to the advantages gained through our innovations. Specifically, the LDP-ecGAN model, a core component of PB-fdGAN, enhances the generation of useable data, effectively addressing the issue of data imbalance previously observed in both the WSTS and AWID datasets. Furthermore, the DFD collaboration scheme is designed in PB-fdGAN enabling multiple industrial CPSs to collectively build a comprehensive intrusion detection model without exchanging their sub-network flows and an isomorphic neural network model. This is complemented by the QoS-HLF framework, which implements a performance-weighted aggregation approach to identify and select the most effective intrusion detection models based on their performance and robustness metrics. Consequently, the efficacy of PB-fdGAN is not only evident in the enhanced performance of intrusion detection but also in the substantial reduction of class imbalance, privacy gap, model homogenization, and arbitrary aggregation.

6.3. Performance comparison

Here, two representatives, i.e., Deep Learning (DL)-GAN (Zhao et al., 2022) and MD-BiGAN (Wang et al., 2022b), are compared with the proposed PB-fdGAN, analyzing their performance on both the WSTS and AWID datasets. They independently follow GAN application and federated learning for IDSs and show very competitive performance in recent literature. For the former, DL-GAN incorporates an advanced discriminator to create practical attack samples that boost detection capabilities. The latter combines BiGAN model with MD-GAN framework to jointly train an efficient intrusion detection model, aiming to improve the classification performance. It should be noted that we do not directly cite the results from Refs (Zhao et al., 2022)-(Wang et al., 2022b); instead, we have implemented these IDS solutions and conducted experiments within our simulations.

Table 5
Comparison of *Precision*, *Recall*, *F1-Score*, and *FAR* of existing IDSs.

IDS Implementation	Precision	Recall	F1-Score	FAR
i) Without any Mechanism				
GHSOM (Liang et al., 2021)	96.17	96.11	95.99	5.48
Hypothesis Testing (Khan et al., 2022)	94.28	94.00	93.96	7.15
Dolphine + SVM (Gao et al., 2022)	95.99	95.97	95.87	5.59
Markov Model (Alohali et al., 2022)	91.33	93.17	92.00	10.40
ii) With Auxiliary Mechanism				
GHSOM + MOEA (Liang et al., 2020)	97.44	97.03	97.01	2.72
t-test + Bayesian (Hao et al., 2021)	96.46	96.18	96.08	5.42
ADMM + ERM (Belenguer et al., 2022)	98.42	98.28	98.30	1.28
Proposed PB-fdGAN	99.00	98.99	98.97	0.79

Fig. 11(a–c) illustrates the comparison results under the WSTS dataset, showing that the *Precision*, *Recall*, and *FAR* of the proposed PB-fdGAN outperforms those of DL-GAN (Zhao et al., 2022) and MD-BiGAN (Wang et al., 2022b), as the training epochs (T) increase from 10 to 50. Similarly, Fig. 11(d) indicates the *F1-Score* of PB-fdGAN surpasses that of the two compared IDS solutions. Apart from that, the comparison results under the AWID dataset are shown in Table 4, which displays the Jaccard coefficients comparisons for three different IDS solutions across various attack types. According to Table 4, the proposed PB-fdGAN demonstrates superior Jaccard coefficients against various attack types compared to the two other IDS solutions, particularly excelling in detecting sparse attack classes such as $J_{flooding}$, $J_{injection}$, and $J_{impersonation}$.

This superior performance of the proposed PB-fdGAN primarily stems from the innovative LDP-ecGAN model, which serves not only as a classifier—utilizing a deep learning network to distinguish between attack and normal flows—but also as a generator, creating synthetic samples to bolster classification of rare classes. Most notably, PB-fdGAN incorporates the DFD collaboration scheme and QoS-HLF framework to comprehensively train the IDS model across the entired industrial CPSs. In contrast, DL-GAN lacks any collaboration mechanism to address issues like data imbalance situations found in datasets such as WSTS and AWID. Meanwhile, the collaboration in MD-BiGAN is limited to a unidirectional flow from the multiple generators and encoders to the central discriminator. Consequently, the proposed PB-fdGAN has better performance than the two representatives in terms of *Precision*, *Recall*, *F1-Score*, and *FAR*.

Additionally, the comparison among the existing IDSs in industrial CPSs is provided in Table 5, which is divided into two parts: the IDSs with and without *auxiliary* mechanisms. According to these comparisons, the proposed IDS solution, i.e., PB-fdGAN, reaches the highest values of *Precision, Recall*, and *F1-Score* while securing the lowest *FAR* compared with other prevailing IDS solutions. Therefore, the performance of our proposed PB-fdGAN has demonstrated superiorities over the state-of-art IDS solutions in industrial CPSs.

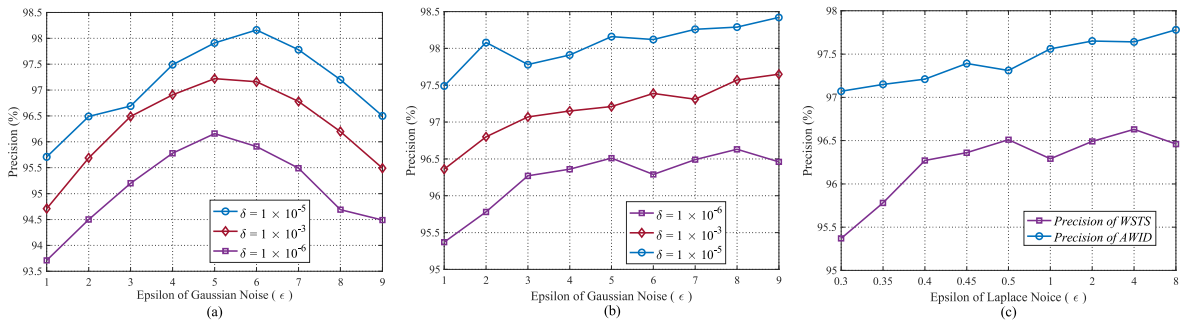

Fig. 12. The Effect of Different Noise on PFD-GAN: (a) Impact of Gaussian Noise under WSTS, (b) Impact of Gaussian Noise under AWID, (c) Impact of Laplace Noise under both WSTS and AWID.

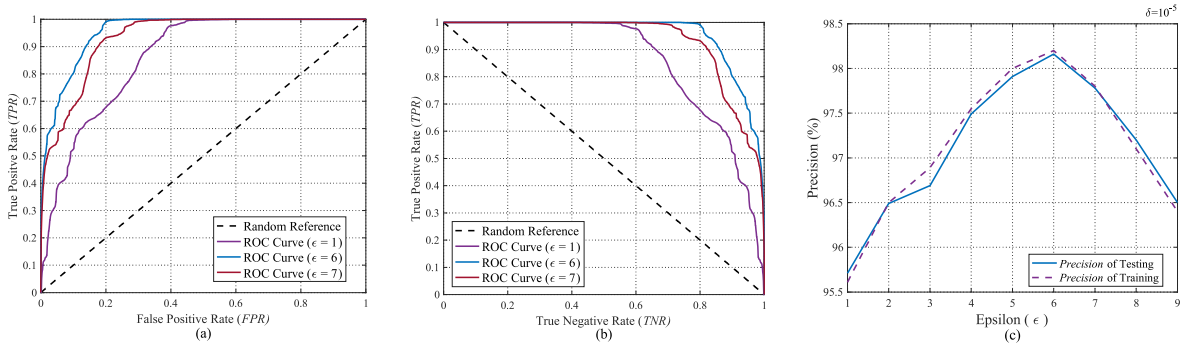

Fig. 13. Detailed analysis of Gaussian noise scales on PB-fdGAN performance: (a) ROC curve, (b) mirror ROC curve, and (c) precision of various noise levels.

6.4. Discuss of important parameters

In order to apply LDP technology to the proposed PB-fdGAN, experiments with the most common Gaussian noise and Laplace noise (Muthukrishnan et al., 2023) types are conducted to investigate the impact of different noises on the performance of the proposed IDS solution. Gaussian noise is imposed with zero mean (no bias) and multiple values of standard deviation σ resulting in (ϵ, δ)-differential privacy, in which σ is calculated by $\sqrt{2\log\left(\frac{1.25}{\delta}\right)}/\epsilon$. The Laplace noise is primarily influenced by the privacy budget ϵ, which represents the degree of privacy leakage. Different types of noise and the noise scales can affect the *Precision*, *Recall*, *F1-Score*, and *FAR* of PB-fdGAN. Take Gassian noise scales as an example, keeping δ fixed (δ is usually set as 10^{-5} and has a tiny impact on the detection performance (Bu et al., 2020)), the *Precision*, *Recall*, and *F1-Score* decrease, and the *FAR* rises if we set an inappropriate ϵ. On the contrary, when ϵ is assigned with a proper value, the *Precision*, *Recall*, and *F1-Score* increase while a lower *FAD* is observed. Hence, it is important to find an appropriate (ϵ, δ) for PB-fdGAN.

The impact of different noise types on the performance of PB-fdGAN is shown in Fig. 12. An analysis of the vertical comparison between Fig. 12(a) and (b) intuitively reveals that when $\delta = 10^{-5}$ Gaussian noise has the least impact on PB-fdGAN, regardless of the dataset used. Furthermore, a horizontal comparison between Fig. 12(a) and (c) indicates that the performance of PB-fdGAN with Gaussian noise scaled by $(\epsilon \sim \{1, 9\}, \delta = 1 \times 10^{-5})$ is superior to that with Laplace noise scaled by $\epsilon \sim \{0.3, 8\}$, making Gaussian noise the more suitable choice for implementing LDP technology.

Upon selecting the Gaussian noise, it is essential to delve into the influence of different noise scales and scale parameters. The curves of Receiver Operation Characteristic (ROC), mirror ROC, and *Precision* for various (ϵ, δ) pairs are recorded to reflect the performance of PB-fdGAN in different Gaussian noise levels, which are created by plotting *True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR)* at various threshold setting, as shown in Fig. 13. In Fig. 13, 3 different vectors, including $(1, 10^{-5})$, $(6, 10^{-5})$, and $(7, 10^{-5})$, are assigned to (ϵ, δ) to manifest the performance of PB-fdGAN. Compare with the other two (ϵ, δ) pairs, the model adopting $(6, 10^{-5})$ as the noise scale gains a bigger Area Under ROC Curve (AUC). Moreover, PB-fdGAN achieves the highest values of *Precision* for $\epsilon = 6$ and $\delta = 10^{-5}$. In other words, the noise scale $(6, 10^{-5})$ is regarded as a suitable vector for (ϵ, δ).

7. Conclusions and future work

In this article, PB-fdGAN is proposed as a secure and private-assured solution for CIDS in industrial CPSs. Firstly, the novel LDP-ecGAN is developed with leveraging Wasserstein distance and label condition to improve the capabilities of generation and intrusion detection, while using LDP technique to prevent privacy leakage. Additionally, the DFD collaboration scheme is designed, enabling multiple LDP-ecGANs of industrial CPSs to collaboratively build a comprehensive intrusion detection model without exchanging sub-network flows and sharing an isomorphic neural network model. Moreover, the QoS-HLF framework is crafted, which introduces an innovative QoS evaluation methodology to select preferred intrusion detection models for performance-weighted aggregation and constructs customized CCs to guarantee the reliable execution of the DFD collaboration against poisoning and membership inference attacks. The in-depth theory analysis is conducted to manifest the security requirements that our PB-fdGAN satisfies, while extensive experimental validation on real-world industrial CPS datasets reveals the superiority of PB-fdGAN over existing state-of-the-art IDS solutions, showcasing significant improvements in Jaccard coefficients, *Precision*, *Recall*, *F1-Score*, and reduction in *FAR*. Notably, PB-fdGAN demonstrates exceptional efficacy in detecting sparse and challenging attack categories, which are often overlooked by conventional methods. The primary advantage of the PB-fdGAN is its applicability in diverse and challenging environments where data privacy and robust intrusion detection are paramount. This versatility makes it especially suitable for other scenarios such as smart grid systems, autonomous vehicular

networks, and IoT devices in smart cities, where similar privacy and security challenges exist. The adaptability of PB-fdGAN to different industrial settings and its ability to maintain high detection accuracy while ensuring data confidentiality suggest its potential as a foundational technology for future resilient CPSs security frameworks.

The scalability and node quantity aspects of this research encounter limitations due to the modest scope of the initial experiments, which are constrained by the availability of resources and aimed primarily at validating the proposed methodologies rather than assessing large-scale system performance. Future research will be directed towards improving model efficiency and scalability, expanding the experimental framework to accommodate a larger number of nodes, and assessing the system's performance under large-scale conditions. Additionally, there is a commitment to enhancing the robustness of the models against evolving cyber threats and refining the QoS evaluation metrics to adapt to the dynamic and complex nature of industrial environments.

CRediT authorship contribution statement

Junwei Liang: Funding acquisition, Formal analysis, Data curation, Conceptualization. **Muhammad Sadiq:** Methodology, Investigation. **Geng Yang:** Methodology. **Kai Jiang:** Validation. **Tie Cai:** Resources, Project administration. **Maode Ma:** Validation, Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This work was supported by the Shenzhen Basic Research Foundation (No. 20220820003203001), the Shenzhen Science and Technology Program (No. RCBS20221008093252092), the Science and Technology Ph.D. Research Startup Project (SZIIT2023KJ016), and the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515110070 and No. 2022A1515110667).

References

Alohali, Manal Abdullah, et al., 2022. Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment. Cognitive Neurodynamics 16 (5), 1045–1057.
Aloqaily, Moayad, et al., 2022. Towards blockchain-based hierarchical federated learning for cyber-physical systems. In: 2022 International Balkan Conference on Communications and Networking (BalkanCom). IEEE, pp. 46–50.
Althobaiti, Maha M., et al., 2021. An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems. Measurement 186, 110145.
Belenguer, Aitor, Navaridas, Javier, Pascual, Jose A., 2022. A Review of Federated Learning in Intrusion Detection Systems for Iot arXiv preprint arXiv:2204.12443.
Bu, Zhiqi, et al., 2020. Deep learning with Gaussian differential privacy. Harvard data science review (23), 10–1162.
Cervini, J., Rubin, A., Watkins, L., 2022. Don't drink the cyber: extrapolating the possibilities of Oldsmar's water treatment cyberattack. In: International Conference on Cyber Warfare and Security. Academic Conferences International Limited, pp. 19–25.
Chatzoglou, Efstratios, Kambourakis, Georgios, Kolias, Constantinos, 2021. Empirical evaluation of attacks against IEEE 802.11 enterprise networks: the AWID3 dataset. IEEE Access 9, 34188–34205.
Cyber-Physical System. [Online]. Available: https://en.wikipedia.org/wiki/Cyber-physical_syst-em, Access on: 2023.
Dong, Jinshuo, Roth, Aaron, Su, Weijie J., 2022. Gaussian differential privacy. J. Roy. Stat. Soc. B Stat. Methodol. 84 (1), 3–37.
Gao, Ying, et al., 2022. Self-learning spatial distribution-based intrusion detection for industrial cyber-physical systems. IEEE Transactions on Computational Social Systems 9 (6), 1693–1702.
George, A.S., Baskar, T., Srikaanth, P.B., 2024. Cyber threats to critical infrastructure: assessing vulnerabilities across key sectors. Partners Universal International Innovation Journal 2 (1), 51–75.
Gui, Jie, et al., 2021. A review on generative adversarial networks: algorithms, theory, and applications. IEEE Trans. Knowl. Data Eng. 35 (4).
Hao, Weijie, Yang, Tao, Yang, Qiang, 2021. Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber-physical systems. IEEE Trans. Autom. Sci. Eng. 20 (1), 32–46.
Haque, Ayaan, 2020. Ec-gan: Low-Sample Classification Using Semi-supervised Algorithms and Gans arXiv preprint arXiv:2012.15864.
Hardy, Corentin, Le Merrer, Erwan, Sericola, Bruno, 2019. Md-gan: multi-discriminator generative adversarial networks for distributed datasets. In: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, pp. 866–877.
Huang, Xianting, et al., 2022. EEFED: personalized federated learning of Execution&Evaluation dual network for CPS intrusion detection. IEEE Trans. Inf. Forensics Secur. 18, 41–56.
Hyperledger Fabric. [Online]. Available: https://www.hyperledger.org/, Access on: 2023.
Industrial Control System (ICS), 2023. Cyber Attack Datasets. https://sites.goog-le.com/a/uah.edu/tommy-morris-uah/ics-data-sets. Access on.
Keshk, Marwa, et al., 2019. An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. IEEE Transactions on Sustainable Computing 6 (1), 66–79.
Khan, Izhar Ahmed, et al., 2021. A privacy-conserving framework based intrusion detection method for detecting and recognizing malicious behaviours in cyber-physical power networks. Appl. Intell. 1–16.
Khan, Izhar Ahmed, et al., 2022. Enhancing IIoT networks protection: a robust security model for attack detection in Internet Industrial Control Systems. Ad Hoc Netw. 134, 102930.
Kumar, Randhir, Tripathi, Rakesh, 2021. DBTP2SF: a deep blockchain-based trustworthy privacy-preserving secured framework in industrial internet of things systems. Transactions on Emerging Telecommunications Technologies 32 (4).
Li, Beibei, et al., 2020a. DeepFed: federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Trans. Ind. Inf. 17 (8), 5615–5624.
Li, Tian, et al., 2020b. Federated learning: challenges, methods, and future directions. IEEE Signal Process. Mag. 37 (3), 50–60.
Liang, Junwei, Ma, Maode, 2020. FS-MOEA: a novel feature selection algorithm for IDSs in vehicular networks. IEEE Trans. Intell. Transport. Syst. 23 (1).
Liang, Junwei, Ma, Maode, Tan, Xu, 2021. Gadqn-ids: a novel self-adaptive ids for vanets based on bayesian game theory and deep reinforcement learning. IEEE Trans. Intell. Transport. Syst. 23 (8), 12724–12737.
Mansour, Romany F., 2022. Artificial intelligence based optimization with deep learning model for blockchain enabled intrusion detection in CPS environment. Sci. Rep. 12 (1), 12937.
Mothukuri, Viraaji, et al., 2021. FabricFL: blockchain-in-the-loop federated learning for trusted decentralized systems. IEEE Syst. J. 16 (3), 3711–3722.
Mourtzis, D., 2023. Digital twin inception in the Era of industrial metaverse. Frontiers in Manufacturing Technology 3, 1155735.
Muthukrishnan, Gokularam, Kalyani, Sheetal, 2023. Grafting Laplace and Gaussian distributions: a new noise mechanism for differential privacy. IEEE Trans. Inf. Forensics Secur. 18, 5359–5374.
Pivoto, Diego GS., et al., 2021. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: a literature review. J. Manuf. Syst. 58, 176–192.
Quincozes, Silvio Ereno, et al., 2021. On the performance of GRASP-based feature selection for CPS intrusion detection. IEEE Transactions on Network and Service Management 19 (1), 614–626.
Rahman, Ziaur, Yi, Xun, Khalil, Ibrahim, 2022. Blockchain based AI-enabled industry 4.0 CPS protection against advanced persistent threat. IEEE Internet Things J. 10 (8), 6769–6778.
Ramadevi, P., et al., 2022. Deep learning based distributed intrusion detection in secure cyber physical systems. Intelligent Automation & Soft Computing 34 (3).
Salau, Babajide A., Rawal, Atul, Rawat, Danda B., 2022. Recent advances in artificial intelligence for wireless internet of things and cyber-physical systems: a comprehensive survey. IEEE Internet Things J. 9 (15), 12916–12930.
Shen, Tao, et al., 2020. Federated Mutual Learning. *arXiv preprint arXiv:2006.16765*.
Tahir, Bushra, Jolfaei, Alireza, Tariq, Muhammad, 2021. Experience-driven attack design and federated-learning-based intrusion detection in industry 4.0. IEEE Trans. Ind. Inf. 18 (9), 6398–6405.
Wang, Zhendong, et al., 2022a. A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning. Expert Syst. Appl. 206, 117671.
Wang, Weili, et al., 2022b. Federated multi-discriminator BiWGAN-GP based collaborative anomaly detection for virtualized network slicing. IEEE Trans. Mobile Comput. 22 (11), 6445–6459.
Wasserstein GAN. [Online]. Available: https://en.wikipedia.org/wiki/Wasserstein_GAN, Access on: 2023.
Xu, Chugui, et al., 2019. GANobfuscator: mitigating information leakage under GAN via differential privacy. IEEE Trans. Inf. Forensics Secur. 14 (9), 2358–2371.
Zhao, Qingling, et al., 2022. CAN bus intrusion detection based on auxiliary classifier GAN and out-of-distribution detection. ACM Trans. Embed. Comput. Syst. 21 (4), 1–30.
Zhu, Tianqing, et al., 2020. More than privacy: applying differential privacy in key areas of artificial intelligence. IEEE Trans. Knowl. Data Eng. 34 (6), 2824–2843.