Distributed Intrusion Detection And Privacy-Preserving Fusion For Industrial SDN

Yong Wang , Fan Jia*, Yi Ren , Ming Wang, and Jun Li

Operation and Maintenance Center, State Grid Digital Technology Holding Co., Ltd., Xiong'an New Area 071708, Hebei Province, P. R. China

* *Corresponding author. E-mail: jiafan_isc@163.com*

Received: July. 21, 2026; Accepted: Aug. 17, 2026

Software-defined networking (SDN) is increasingly deployed in industrial networks such as industrial Internet of Things and cloud-edge control systems, where centralized controllers, open southbound interfaces, and dynamic flow-table mechanisms enlarge the attack surface. Traditional centralized intrusion detection systems require raw-traffic uploads, impose heavy controller load, risk cross-domain privacy leakage, and generalize poorly under non-IID traffic across industrial subdomains. This paper proposes a scalable distributed intrusion detection and privacy-preserving data fusion mechanism for SDN-based industrial networks. The system deploys local SDN controllers, edge intrusion detection agents, a secure aggregation module, and a global federated fusion server, fusing data-plane traffic, control-plane event, and temporal behavior features through federated learning, secure aggregation, and differential privacy perturbation for cross-domain collaborative detection without exposing raw industrial traffic. Simulations using InSDN, CICIDS2017, and Mininet-generated industrial SDN attack traffic show accuracy of 0.958, an F1-score of 0.949, and an AUC of 0.974 in multi-class classification. As SDN domains grow from 5 to 50, the F1-score decreases only from 0.953 to 0.941 while detection latency rises from 18.4 ms to 41.6 ms. At a privacy budget of 2.0, the success rate of membership inference attacks falls to 0.587, a 28.1% reduction relative to the non-differential-privacy setting. These results show that the method enables scalable, low-overhead, interpretable collaborative intrusion detection for industrial SDN networks without sharing raw traffic, offering a practical solution for privacy-preserving security monitoring in large-scale industrial information systems.

Keywords: federated learning; differential privacy; secure aggregation; scalable information systems; membership inference attack.

© The Author('s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202612_35.048

1. Introduction

Software-defined networking abstracts control logic from traditional network devices, enabling centralized network control, policy orchestration, traffic scheduling, and security response. As industrial Internet platforms, smart grid substations, cloud-edge industrial control systems, and multi-tenant industrial enterprise networks continue to adopt SDN, industrial SDN networks have evolved from small experimental networks managed by a single controller into scalable distributed information systems spanning multiple domains, controllers, switches, and service slices. Compared with traditional industrial control networks, SDN provides a stronger global view, policy automation, and flow-rule scheduling, but its centralized control, open interfaces, and frequent control-plane interactions also expand the potential impact of attacks on safety-critical industrial infrastructure. Recent stud-

ies show that intrusion detection for industrial SDN has gradually moved from rule-based and traditional machine learning methods toward deep learning, distributed collaborative detection, and privacy-preserving training [1, 2].

In SDN environments, attacks such as DDoS, port scanning, flow-table overflow, controller saturation, and lateral movement not only appear as abnormal data-plane traffic but also trigger abrupt changes in control-plane events. For example, DDoS attacks can rapidly increase packet-in messages, flow-table overflow attacks can raise the table-miss ratio of switches, and controller saturation attacks can increase response delays and CPU usage. Traditional centralized IDSs usually require raw traffic and logs to be uploaded to a central node for training or detection. This model creates three problems in large-scale SDN scenarios. First, raw-traffic transmission consumes bandwidth and increases controller burden. Second, cross-domain sharing of IP addresses, ports, access sequences, and security logs introduces privacy leakage risks. Third, traffic distributions differ markedly across SDN domains, so centralized or single-domain models are prone to degraded generalization under non-IID conditions. Federated learning has been used to alleviate data silos and privacy-sharing problems in network intrusion detection, but ordinary federated IDSs may still leak information through model updates and make insufficient use of SDN-specific control-plane features [3, 4].

Existing research has made progress in deep-learning-based SDN intrusion detection, hybrid feature selection, lightweight traffic classification, and control-plane security. For example, CNN-BiLSTM and other deep learning models can improve network intrusion detection performance, and SDN control capabilities have also been introduced into industrial IoT scenarios for security monitoring [5]. However, most studies still focus mainly on detection accuracy, with insufficient attention to scalable system architecture, cross-domain privacy fusion, controller load, communication overhead, and dynamic attack response. For scalable distributed information systems such as industrial SDN networks, it is not enough to show only high model accuracy. It is also necessary to demonstrate stable system performance as node scale increases, traffic distributions become heterogeneous, attack intensity grows, and privacy constraints are imposed.

Taken together, these studies share three combined limitations: control-plane events are rarely fused with data-plane traffic in a principled way, cross-domain non-IID adaptability is seldom evaluated, and privacy protection is often incomplete, since federated updates may still be exposed to inference or reconstruction attacks. These gaps motivate the four contributions below: Contribution 1 supplies the missing system-level evaluation; Contribution 2 targets insufficient control-plane feature utilization; Contribution 3 targets incomplete privacy protection; and Contribution 4 targets the shortage of non-IID adaptability analysis through explicit domain-scaling and non-IID experiments.

To address these issues, this paper proposes a scalable distributed intrusion detection and privacy-preserving data fusion mechanism for SDN-based industrial networks. It models a multi-domain industrial SDN network as a distributed information system, deploys edge IDS agents alongside local controllers, and constructs a multi-source feature space from data-plane traffic features, control-plane event features, and temporal behavior features. During training, federated learning supports cross-domain model collaboration, while secure aggregation and differential privacy perturbation reduce the risk of model-update leakage [6]. During detection, multi-source attention fusion outputs attack categories and risk scores, and the SDN controller issues rate-limiting, isolation, blocking, and rerouting policies. The main contributions of this paper are as follows:

First, the paper proposes a scalable distributed intrusion detection framework for multidomain industrial SDN, deploying traffic collection, local detection, privacy fusion, and global response in layers to reduce the controller and bandwidth pressure caused by centralized IDSs. Second, it constructs a feature-fusion mechanism for data-plane, control-plane, and temporal behavior features, incorporating SDN-specific indicators such as packet-in frequency, flowtable occupancy, table-miss ratio, and controller response latency into the detection model [7]. Third, it designs a privacy-preserving training mechanism that combines federated learning, secure aggregation, and differential privacy to achieve cross-domain collaborative detection without sharing raw traffic, IP addresses, ports, or session logs. Fourth, it conducts in-depth simulations covering algorithm performance, attack-type identification, SDN domain scaling, topology robustness, attack intensity, controller load, communication overhead, privacy budget, non-IID traffic, dynamic attack evolution, and ablation settings.

2. Methods

2.1. System Model and Threat Modeling

2.1.1. Multi-domain SDN Network Model

This paper considers a large-scale industrial network environment consisting of multiple SDN subdomains. Each

subdomain contains OpenFlow switches, end hosts, a local SDN controller, and an edge IDS agent. Different subdomains may belong to different industrial control areas, departments, campuses, or data-center partitions. Because of business isolation, privacy compliance, and management-access restrictions, subdomains cannot directly share raw traffic packets, IP addresses, port numbers, or access-session logs; instead, they collaborate through protected model updates or de-identified risk summaries. The global federated fusion server does not access raw data and is responsible only for receiving locally encrypted or perturbed model updates and aggregating them.

To demonstrate scalability, this paper sets the number of SDN domains to 5, 10, 20, 30, and 50 , with each domain containing 4 to 12 virtual switches and 20 to 100 hosts. The simulation topologies include Tree, Fat-tree, Random, and Hybrid structures. The Tree topology simulates hierarchical campus networks, the Fat-tree topology simulates data-center networks, the Random topology simulates complex enterprise network connections, and the Hybrid topology simulates cloud-edge collaboration and multi-service domain deployments. This setup covers the effects of controller load, switch scale, host scale, and path changes on the intrusion detection system.

2.1.2. *Intrusion Threat Model*

This paper focuses on five typical SDN attacks. The first is DDoS, in which attackers trigger switches to send packet-in messages to the controller through large volumes of forged or high-frequency traffic, causing control-plane congestion [8]. The second is port scanning, in which an attacking host accesses many ports within a short period, increasing destination-port entropy and connection failure rates. The third is flow-table overflow, in which attackers create many short-lived flows that fill switch flow tables with abnormal rules, increasing table-miss rates and rule-installation latency. The fourth is controller saturation, in which attack traffic or abnormal control messages significantly increase controller CPU utilization and response latency. The fifth is lateral movement, in which compromised hosts perform lateral access among service nodes, causing frequent path changes and abnormal communication sequences.

Attackers are assumed to be able to launch external traffic attacks and internal lateral movement, but not to fully control the global fusion server. For privacy attacks, this paper considers honest-but-curious global servers and passive eavesdroppers that may attempt to infer local sample membership or reconstruct traffic features from single-domain model updates. Therefore, the system must reduce the risk of membership inference attacks and model inversion while maintaining detection performance. Related work suggests that control-plane overload and intrusion detection overhead in SDN scenarios should be evaluated together with detection accuracy and system operating costs [9].

2.1.3. *Privacy Constraints and Optimization Goals*

Let the local dataset of the k-th SDN domain be $D_k = \{(x_{k,i}, y_{k,i})\}_{i=1}{}^{n}k$, where $x_{k,i}$ represents multi-source network features and $y_{k,i}$ represents traffic category or attack type. Due to privacy constraints, D_k cannot be transmitted to other SDN domains or the global server. The system allows only protected local model updates $\Delta\theta_k$. The global objective is to minimize cross-domain empirical risk without exposing the original data.

$$\min_{\theta} F(\theta) = \sum_{k=1}^{k} \frac{n_k}{n} l_k(\theta), \quad n = \sum_{k=1}^{k} n_k \tag{1}$$

Here, K represents the number of SDN domains, θ represents the global detection model parameters, and $L_k(\theta)$ represents the local detection loss on the k-th SDN domain. In addition to detection accuracy, the system also needs to minimize detection latency T_{det} , controller response latency T_{ctrl} , communication overhead C_{comm} , and privacy leakage risk R_{priv} . Therefore, the study adopts a multi-objective evaluation approach that incorporates detection performance, system scalability, and privacy-preservation effectiveness into the simulation analysis.

2.2. Scalable Distributed Intrusion Detection and Privacy Fusion Framework

2.2.1. *Overall Architecture*

The scalable distributed intrusion detection framework proposed in this paper is shown in Fig. 1. The system consists of the SDN data plane, local controllers, edge IDS agents, a privacy-preserving fusion module, and a global federated fusion server [10]. In the data plane, OpenFlow switches forward traffic and record flow-table hits, table misses, and packet-in events. Local controllers install flow rules and distribute security policies. Edge IDS agents are deployed beside controllers and perform local data collection, feature extraction, attack detection, and local model training. The privacy-preserving fusion module applies gradient norm clipping, differential privacy perturbation, and secure aggregation to model updates, while the global federated fusion server generates a global model and sends it back to each SDN domain.

Fig. 1 shows that the system does not centralize raw traffic for single-point analysis; instead, it performs local detection of traffic and control events on the local controller side. This reduces cross-domain raw-traffic transmission and lowers the pressure on controllers and core links [11].

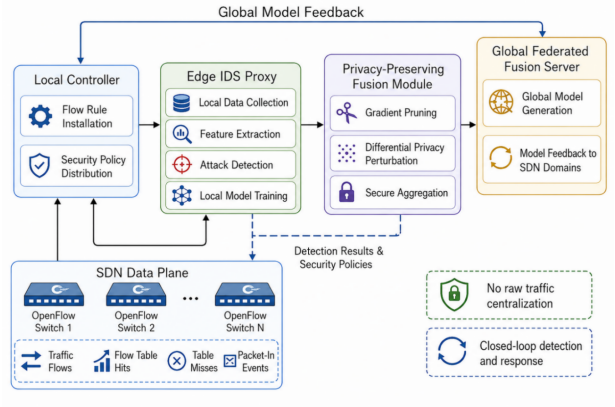


Fig. 1. Scalable distributed IDS and privacy-preserving fusion framework for SDN

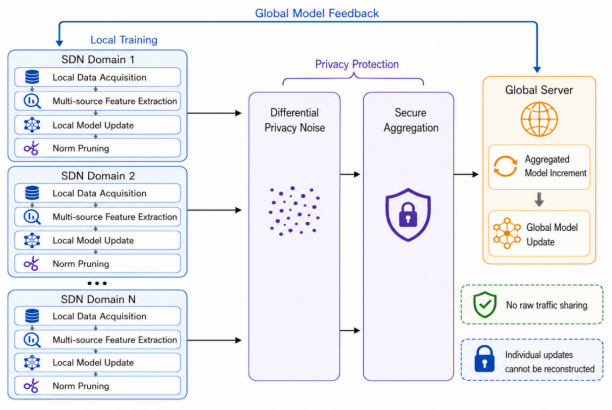


Fig. 2. Privacy-preserving federated data fusion workflow

Compared with traditional centralized IDSs, the architecture is more suitable for scenarios with multiple controllers, subdomains, and services. Compared with ordinary federated IDSs, this paper further incorporates SDN control-plane events into feature fusion and feeds detection results back to the controller policy layer, forming a closed loop of detection and response.

2.2.2. *Edge IDS Agent and Security Response Mechanism*

The edge IDS agent is the key execution unit of the system. Its inputs include switch traffic statistics, local controller event logs, and host access sequences. Within each sliding time window, the agent constructs multi-source feature vectors and performs local model inference. When the risk score exceeds a threshold, the agent sends policy recommendations to the controller. The controller applies different responses according to risk level: low-risk traffic is logged, medium-risk traffic receives increased sampling, high-risk traffic is rate-limited and isolated, and severe-risk traffic is blocked or rerouted. This design makes the IDS part of the SDN security control loop, rather than merely an offline classifier.

2.2.3. *Privacy-Preserving Federated Fusion Process*

The privacy-preserving fusion process is shown in Fig. 2. Each SDN domain first completes local data collection and multi-source feature extraction, and then updates the detection model using local data [12]. To prevent model updates from leaking local traffic distributions, the system clips the local gradient norm, adds differential privacy noise, and hides individual-domain updates through secure aggregation. The global server obtains only the aggregated model increment and cannot directly reconstruct the independent update of any SDN domain.

In the r-th round of federated training, the local model update is denoted as $\Delta\theta_k{}^r$. To limit the impact of a single update on the global model, this paper employs L_2 norm clipping:

$$\Delta^{-}\theta_k^r = \frac{\Delta\theta_k^r}{\max\left(1, \frac{\left\|\Delta\theta_k^r\right\|_2}{C}\right)} \tag{2}$$

Here, C is the clipping threshold. Gaussian noise is then added to generate differentially private updates:

$$\tilde{\Delta}\theta_k^r = \Delta^{-}\theta_k^r + N\left(0, \sigma^2 C^2 \mathrm{I}\right) \tag{3}$$

The global update after secure aggregation is as follows:

$$\theta^{\mathrm{r}+1} = \theta^r + \sum_{k=1}^{k} \frac{n_k}{n} \tilde{\Delta}\theta_k^r \tag{4}$$

This process reduces the likelihood that individual-domain data can be inferred while preserving cross-domain collaborative learning. Compared with using only differential privacy, secure aggregation further lowers the risk that the global server can directly observe local updates. Compared with using only secure aggregation, differential privacy improves protection against membership inference and model inversion attacks [13].

2.3. Multi-Source Feature Fusion and Intrusion Detection Model

2.3.1. *Data-Plane, Control-Plane, and Temporal Behavior Features*

The feature system in this paper includes three categories. The first category is data-plane traffic features, including packet rate, byte rate, flow duration, average packet size, TCP flag ratio, source IP entropy, destination-port entropy, and flow inter-arrival time. These features describe external traffic changes, port probing, and traffic bursts. The second

category is control-plane event features, including packet-in frequency, flow-mod frequency, table-miss rate, flow-table occupancy, rule-installation delay, controller response delay, controller CPU utilization, and switch-controller message rate. These features are a key basis for distinguishing SDN intrusion detection from ordinary network IDS and are especially useful for identifying flow-table overflow and controller saturation attacks [14]. The third category is temporal behavior features, including host access sequences, path-switching frequency, burst traffic patterns, connection growth rate, abnormal communication intervals, and sliding-window flow variation. These features help identify lateral movement and low-rate scanning.

Let the data-plane, control-plane, and temporal behavior features of sample i in time window t be denoted by ${x_{i,t}}^{\text{tra}}$, ${x_{i,t}}^{\text{ctrl}}$, and ${x_{i,t}}^{\text{temp}}$, respectively. Then the fused input is represented as:

$$\mathrm{X}_{i,t} = \left\lfloor \mathrm{X}_{i,t}^{\text{tra}} ; \mathrm{X}_{i,t}^{\text{ctrl}} ; \mathrm{X}_{i,t}^{\text{temp}} \right\rfloor \tag{5}$$

To avoid training bias caused by differences in feature scale, this paper standardizes continuous variables:

$$\hat{x}_j = \frac{x_j - \mu_j}{\sigma_j + \varepsilon_0} \tag{6}$$

Here, μ_j and σ_j represent the mean and standard deviation of the j th feature in the local training set, respectively, and ε_0 is a smoothing term used to prevent division by zero.

2.3.2. *Multi-Source Attention Fusion Model*

The detection model used in this paper is called SPF-IDS (Scalable Privacy-Preserving Fusion IDS). The model consists of three encoders, a multi-source attention fusion layer, an attack classification head, and a risk scoring head [15]. The three encoders process traffic statistics features, control plane events, and temporal behavior features, respectively, to obtain hidden representations ${h_{i,t}}^{\text{tra}}$, ${h_{i,t}}^{\text{ctrl}}$ and ${h_{i,t}}^{\text{temp}}$. For the m-th feature source, the attention score is defined as:

$$e_{i,t}^{(m)} = \mathrm{V}^{\mathrm{T}} \tanh\left(\mathrm{W}_m \, \mathrm{h}_{i,t}^{(m)} + \mathrm{b}_m \right), \quad m \in \{ \text{ tra , ctrl, temp } \} \tag{7}$$

The normalized attention weights are:

$$a_{i,t}^{(m)} = \frac{\exp\left(e_{i,t}^{(m)}\right)}{\sum_{q \in \{ \text{ tra , ctrl , temp } \}} \exp\left(e_{i,t}^{(q)}\right)} \tag{8}$$

The final fusion is represented as:

$$\mathrm{Z}_{i,t} = \sum_{m \in \{ \text{ tra , ctrl , temp } \}} \mathrm{a}_{i,t}^{(m)} \mathrm{h}_{i,t}^{(m)} \tag{9}$$

The attack classification probability is:

$$\hat{y}_{i,t} = \mathrm{softmax}\left(\mathrm{W}_c \mathrm{z}_{i,t} + \mathrm{b}_c\right) \tag{10}$$

To address attack-class imbalance, the local loss function uses class-weighted crossentropy with a regularization term:

$$L_k(\theta) = -\frac{1}{|D_k|} \sum_{(x,y) \in D_k} \sum_{c=1}^{C_a} w_c \Pi(y = c) \log \hat{y}_c + \lambda \|\theta\|_2^2 \tag{11}$$

Here, C_a represents the number of attack categories, w_c represents the category weight, and λ is the regularization coefficient. Risk scoring relies not only on classification probabilities but also on the anomaly strength of the SDN control plane.

$$\begin{aligned} R_{i,t} = \beta_1 \max_{c \neq \text{normal}} \hat{y}_{i,t,c} + \beta_2 \mathrm{Z}_{pktin} \\ + \beta_3 \mathrm{Z}_{miss} + \beta_4 \mathrm{Z}_{delay} + \beta_5 \mathrm{Z}_{cpu} \end{aligned} \tag{12}$$

Here, z_{pktin}, z_{miss}, z_{delay} and z_{cpu} represent the standardized anomaly scores for packet-in frequency, table-miss rate, controller response latency, and CPU utilization, respectively. This risk score can be used directly for tiered SDN security responses.

2.4. Privacy-Preserving Federated Training Mechanism

2.4.1. *Local Training and Global Aggregation*

Each SDN domain performs several rounds of mini-batch training locally. Let the k-th domain receive the global model θ^r in round r. After local training for E epochs, it obtains the ${\theta_k}^{r,E}$ and defines the model increment as:

$$\Delta\theta_k^r = \theta_k^{r,E} - \theta^r \tag{13}$$

After norm clipping, differential privacy perturbation, and secure aggregation, local increments are used in the global update. Because sample sizes differ across SDN domains, the aggregation weights are adjusted as n_k / n. This weighting strategy reduces the random perturbation introduced by small-sample domains while preserving each domain's contribution to specific attack types. For non-IID traffic, this paper introduces a proximal constraint into the local loss to stabilize model updates:

$$L_k^{\text{prox}}(\theta) = L_k(\theta) + \frac{\mu}{2} \|\theta - \theta^r\|_2^2 \tag{14}$$

Here, μ is the proximal regularization coefficient, which mitigates model drift caused by distribution differences across SDN domains.

2.4.2. *Privacy Protection and Security Analysis*

Differential privacy limits the influence of individual samples on the final model by injecting noise into model updates after norm clipping. Given privacy budget ϵ and failure probability δ, the noise scale σ is inversely related to ϵ. A smaller ϵ provides stronger privacy protection but lowers detection accuracy; a larger ϵ can improve model performance but increases membership inference risk [16]. Based on the privacy-budget sensitivity experiment, this paper selects $\epsilon = 2.0$ as the main setting to balance detection performance and privacy protection. Existing research on federated intrusion detection also identifies privacy protection, communication efficiency, and non-IID robustness as core implementation challenges for federated IDSs.

The secure aggregation module uses a random mask to hide local updates. For the k-th domain, the locally uploaded value can be represented as:

$$s_k = \tilde{\Delta}\theta_k + \sum_{j>k} r_{k,j} - \sum_{j<k} r_{j,k} \tag{15}$$

Here, $r_{k,j}$ is a random mask shared by domains k and j. After the uploaded values from all domains are summed, the paired masks cancel each other out.

$$\sum_{k=1}^{k} s_k = \sum_{k=1}^{k} \tilde{\Delta}\theta_k \tag{16}$$

Therefore, the global server obtains only the aggregated result and cannot directly observe individual-domain model updates. This mechanism is suitable for collaborative security detection in multi-organization, multi-service, and multi-tenant SDN networks.

This offers a concrete anti-model-inversion advantage over plaintext uploading: a gradient-inversion attack, which optimizes a synthetic input to match an observed plaintext update and can reconstruct input features with low error for small batches, cannot be mounted on any single domain's masked value, since it is statistically indistinguishable from noise to a non-colluding server and only the sum across all domains is ever exposed. Ablation results in Table 8 quantify the resulting cost: removing secure aggregation raises accuracy only from 0.958 to 0.961 and F1-score only from 0.949 to 0.952 , showing that this protection closes a plaintext-inversion channel at little cost to detection performance, complementing the purely statistical guarantee of differential privacy.

3. Results and discussion

3.1. Experimental Environment and Dataset

The experiment combined publicly available intrusion detection datasets with Mininetgenerated industrial SDN traffic. Public datasets were used to validate the model's general intrusion detection capability, while the simulated traffic was used to generate SDN controlplane metrics, including packet-in frequency, flow-table occupancy, table-miss rate, ruleinstallation latency, and controller response latency. The simulation platform used Ubuntu 20.04, Python 3.9, PyTorch, Mininet, the Ryu controller, Open vSwitch, and OpenFlow 1.3. The federated training setting used 100 global epochs, 3-5 local epochs, a batch size of 64, the Adam optimizer, and a learning rate of 0.001. Attack-traffic ratios were set to 5%, 10%, 20%, 30%, and 40%, and detection windows were set to 1 s, 3 s, 5 s, and 10 s.

Table 1 maps attack types to SDN control-plane observation metrics, avoiding the oversimplification of SDN intrusion detection as a routine traffic classification task. Based on this threat model, the subsequent experiments evaluate detection performance, system scalability, privacy-preservation effectiveness, and interpretability.

3.2. Overall Detection Performance Comparison

This section uses Fig. 3 to compare the accuracy, F1-score, and AUC of different methods, including SVM-IDS, RF-IDS, CNN-IDS, LSTM-IDS, FL-IDS, DP-FL-IDS, and the proposed SPF-IDS. To keep the layout clear, this section presents the simulation results as figures rather than adding separate tables.

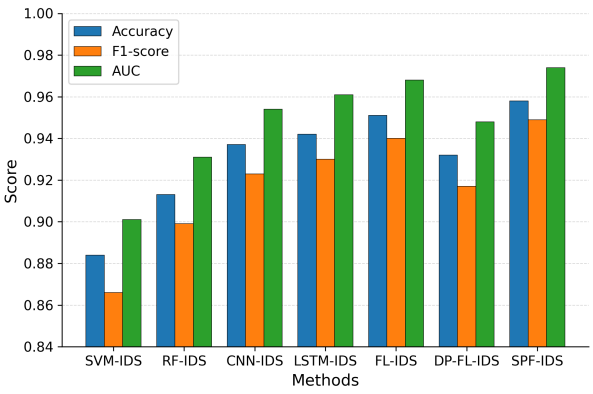


Fig. 3. Detection performance comparison of different IDS methods

As shown in Fig. 3, traditional SVM and RF methods achieve some detection capability with multi-source SDN features, but they struggle to model the complex nonlinear relationships between control-plane events and temporal behaviors. CNN and LSTM models show clear performance improvements, indicating that deep models can learn more effective traffic representations [17]. Ordinary FL-IDS achieves an F1-score of 0.940 without sharing raw

Table 1. Threat model and key observations in SDN networks

Attack type	SDN impact	Key observation metrics	Response strategy
DDoS	High-volume abnormal flows trigger the controller	packet-in rate, source entropy	Rate limiting and blocking
Port Scan	Rapid multi-port probing	port entropy, failed connection ratio	Increase monitoring and isolate hosts
Flow Table Overflow	Flow tables are filled with abnormal rules	table occupancy, table miss rate	Clean rules and reroute
Controller Saturation	Controller processing capacity decreases	CPU utilization, response delay	Protect and downsample
Lateral Movement	Lateral access to internal nodes	path switching, access sequence	Host isolation and path auditing

data, but its robustness remains limited because it lacks privacy perturbation and control-plane fusion. Adding differential privacy reduces the F1-score of DP-FL-IDS to 0.917. The proposed SPF-IDS compensates for information loss caused by privacy noise through multi-source attention fusion, achieving an accuracy of 0.958, an F1-score of 0.949, and an AUC of 0.974, approaching centralized-model performance without uploading raw traffic. These results show that distributed privacy-preserving mechanisms do not necessarily sacrifice detection performance; the key is to make full use of SDN-specific features.

3.3. Detection Performance for Different Attack Types

Table 2 shows that the proposed method performs well in identifying DDoS and flowtable-overflow attacks. This is because both attacks cause abrupt changes in the packet-in rate, flow-table occupancy, and table-miss rate. The relatively lower F1-scores for controller saturation and lateral movement are mainly because these attacks do not necessarily generate large traffic spikes in their early stages; instead, they rely more on controller response latency, path-switching frequency, and changes in access sequences [18]. This result demonstrates the importance of temporal behavior features and control-plane features for detecting sophisticated SDN attacks.

Table 2. Detection performance under different attack types

Attack type	Precision	Recall	F1 score	FPR
Normal	0.973	0.981	0.977	0.018
DDoS	0.965	0.958	0.961	0.022
Port Scan	0.947	0.936	0.941	0.031
Flow Table Overflow	0.952	0.944	0.948	0.027
Controller Saturation	0.938	0.927	0.932	0.039
Lateral Movement	0.921	0.908	0.914	0.046

3.4. SDN Domain Scalability Analysis

Fig. 4 evaluates changes in detection performance and latency as the number of SDN domains expands from 5 to 50. As shown in Fig. 4, when the number of SDN domains increases from 5 to 50, the F1-score decreases from 0.953 to 0.941 , a decrease of 1.2 percentage points, while average detection latency increases from 18.4 ms to 41.6 ms. This change indicates that communication synchronization, model aggregation, and local detection pressure do increase as the system scales, but performance degradation is gradual and latency remains within an acceptable range [19]. Compared with centralized IDSs, SPF-IDS transmits only model updates rather than raw traffic; therefore, its scaling bottleneck mainly comes from federated synchronization rather than centralized traffic upload. This experiment directly addresses the scalability evaluation requirements of scalable information systems.

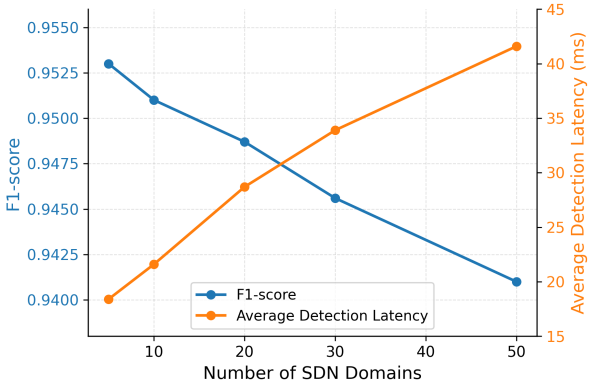


Fig. 4. Scalability analysis under different SDN domain sizes

3.5. Robustness Analysis of Different Network Topologies

Table 3 shows that the model performs best under the Fat-tree topology because datacenter topologies have more regular paths and more stable traffic patterns. Under the Random topology, the F1-score drops to 0.936 and controller load rises to 57.9%, indicating that path changes and complex communication relationships make lateral movement and scanning behavior more difficult to identify [20]. The Hybrid topology falls between the two. These results suggest that intrusion detection systems should consider both traffic distribution and control-plane pressure under different topologies, rather than reporting performance on only one topology.

Table 3. Robustness under different SDN topologies

Topology	Accuracy	F1 score	Average latency / ms	Controller load /%
Tree	0.956	0.947	27.8	51.6
Fat-tree	0.961	0.952	25.4	48.2
Random	0.948	0.936	34.7	57.9
Hybrid	0.953	0.943	31.2	54.3

3.6. Controller Load and Attention Allocation under Varying Attack Intensity

This section uses Fig. 5 to evaluate controller CPU utilization under different attackflow ratios and to show the model's attention allocation to multi-source features under different attack types. DDoS samples have the highest weights on packet-rate and packet-in features, indicating that the model captures control-plane message bursts caused by the attack. Port-scan samples have the highest weights on port entropy, flow-table-overflow samples on table occupancy and table misses, controller-saturation samples on controller delay and CPU utilization, and lateral-movement samples on path switching. This figure not only provides interpretability but also shows that SPF-IDS can dynamically focus on different information sources according to attack type, rather than relying on a single traffic statistic [21].

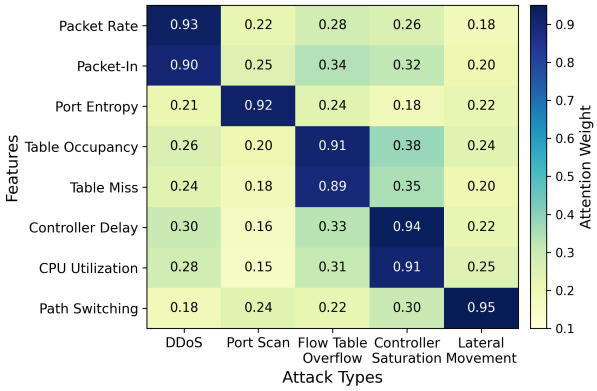


Fig. 5. Attention heatmap of multi-source SDN features under different attacks

3.7. Communication Overhead and Convergence Efficiency

Table 4 shows that centralized IDS achieves high performance in one-time training, but it requires raw traffic and logs to be uploaded, resulting in a single-round communication overhead of 842.5 MB. This makes it unsuitable for cross-domain privacy-sensitive scenarios [22]. Ordinary FL-IDS avoids raw-data upload, but parameter communication overhead remains high. SPF-IDS reduces single-round communication overhead to 96.7 MB through local feature encoding and model-update compression, converging in 64 rounds with a final F1-score of 0.949. This result shows that multi-source fusion does not significantly increase communication burden; instead, it improves convergence efficiency.

Fig. 6 compares the convergence processes of FL-IDS, DP-FL-IDS, FedProx-IDS, and the proposed SPF-IDS in the same multi-domain SDN simulation environment. The horizontal axis represents communication rounds, and the vertical axis represents validation-set F1-score. FL-IDS gradually converges after approximately 78 rounds. DP-FL-IDS converges slowest because differential privacy noise weakens the stability of local model updates, stabilizing after approximately 86 rounds. FedProx-IDS alleviates some training oscillations caused by non-IID traffic, but its final F1-score remains lower than that of the proposed method. In contrast, SPF-IDS improves more rapidly during the first 40 rounds and reaches a stable state after approximately 64 rounds, ultimately achieving an F1-score of 0.949 .

This shows that SPF-IDS does not merely perform federated parameter averaging. Instead, it introduces multi-source SDN feature fusion before local training and incorporates norm clipping, privacy perturbation, and secure aggregation during upload, reducing gradient fluctuations caused by non-IID traffic and privacy noise. As a result, SPF-IDS exhibits a smoother convergence curve and achieves higher stable detection performance with fewer communication rounds [23]. This result further shows that the proposed method improves not only intrusion detection performance but also communication and convergence efficiency in large-scale distributed SDN training scenarios.

3.8. Privacy Budget Sensitivity Analysis

Table 5 shows that a smaller privacy budget provides stronger privacy protection but causes a more pronounced decline in detection performance. When the privacy budget

Table 4. Communication overhead and convergence efficiency

Method	Single-round communication overhead /MB	Convergence rounds	Final F1 score	Raw data shared?
Centralized IDS	842.5	1	0.961	Yes
FL-IDS	126.8	78	0.940	No
DP-FL-IDS	128.4	86	0.917	No
FedProx-IDS	132.7	72	0.934	No
SPF-IDS	96.7	64	0.949	No

Table 5. Effect of privacy budget on detection and privacy protection

Privacy Budget ϵ	Accuracy	F1-score	Membership inference attack success rate	Model inversion risk
0.5	0.918	0.907	0.523	0.218
1.0	0.934	0.923	0.548	0.247
2.0	0.958	0.949	0.587	0.286
4.0	0.963	0.953	0.641	0.339
8.0	0.966	0.957	0.704	0.418
No DP	0.969	0.960	0.816	0.573

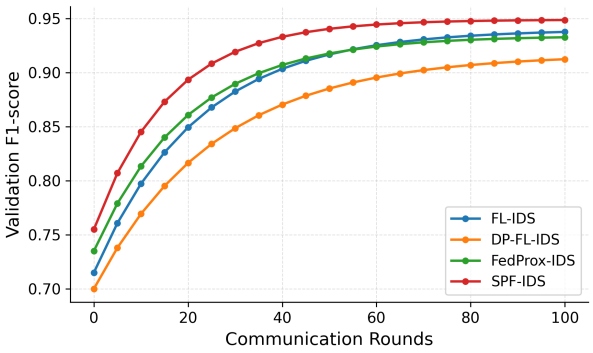


Fig. 6. Convergence efficiency comparison of different federated IDS methods

is 0.5, the membership inference attack success rate drops to 0.523, but the F1-score is only 0.907. When the privacy budget is 8.0 , the F1-score increases to 0.957 , but the membership inference attack success rate rises to 0.704 . At a privacy budget of 2.0, the F1-score is 0.949 and the membership inference attack success rate is 0.587 , achieving a good balance between privacy protection and detection performance. This result is consistent with previous findings on the trade-off between privacy budget and performance in privacy-preserving federated intrusion detection [24, 25].

Table 6. Performance under non-IID traffic distribution

Non-IID level	FL-IDS F1	DP-FL-IDS F1	SPF-IDS F1
Low	0.945	0.924	0.952
Medium	0.927	0.906	0.941
High	0.891	0.872	0.918

3.9. Non-IID Flow Distribution Experiment

Table 6 shows that all methods decline as the non-IID level increases, but SPF-IDS has the smallest decrease. Under high non-IID conditions, the F1-score of ordinary FL-IDS drops to 0.891 and that of DP-FL-IDS drops to 0.872 , while SPF-IDS remains at 0.918 . This indicates that multi-source feature fusion, proximal constraints, and risk correction can mitigate model drift caused by inconsistent attack distributions across SDN domains. This capability is practically important for hybrid deployments involving enterprise networks, data centers, and industrial SDN.

3.10. Dynamic Attack Evolution Experiment

Table 7 shows that the proposed method can track risk continuously as attacks evolve.

The risk score is 0.126 in the normal phase, rises to 0.684 after the port-scan phase begins, and increases further during the DDoS, flow-table-overflow, and controller-saturation phases.

Detection latency and response time increase with attack complexity but remain within the millisecond range. This experiment demonstrates that SPF-IDS is not only a static classification model but can also support continuous security situational awareness for SDN controllers.

Table 7. Dynamic attack evolution and system response

Stage	Time window	Attack status	Average Risk Score	Detection latency / ms	Response time / ms
Stage 1	$0-60$ s	Normal	0.126	15.8	-
Stage 2	$60-120$ s	Port Scan	0.684	24.3	41.6
Stage 3	$120-180$ s	DDoS	0.817	28.7	46.2
Stage 4	$180-240$ s	Flow Table Overflow	0.853	31.5	52.8
Stage 5	$240-300$ s	Controller Saturation	0.891	37.9	61.4

Table 8. Ablation study of SPF-IDS

Model variant	Accuracy	F1-score	Detection latency / ms
Without traffic features	0.931	0.920	31.6
Without control-plane features	0.934	0.926	31.8
Without temporal features	0.941	0.932	29.7
Without attention fusion	0.929	0.918	32.4
Without secure aggregation	0.961	0.952	27.9
Without DP	0.966	0.957	27.6
Full SPF-IDS	0.958	0.949	28.9

Table 8 shows that the F1-score drops to 0.918 after attention fusion is removed, indicating that adaptive multi-source fusion is central to the performance gain. Removing control-plane features reduces the F1-score to 0.926 and has a significant impact on flow-tableoverflow and controller-saturation detection. Removing differential privacy or secure aggregation yields slightly higher detection performance but weakens privacy protection. Therefore, although the full SPF-IDS sacrifices a small amount of accuracy, it achieves a more practical balance among detection performance, privacy protection, and deployment feasibility.

4. Conclusions

This paper addresses the challenges of cross-domain intrusion detection and privacy-preserving data fusion in large-scale SDN-based industrial networks by proposing SPF-IDS, a scalable distributed intrusion detection mechanism. SPF-IDS supports collaborative security detection without sharing raw traffic through edge IDS agents, local controller monitoring, multi-source feature fusion, federated learning, secure aggregation, and differential privacy perturbation. Experimental results show that SPF-IDS achieves an accuracy of 0.958 , an F1-score of 0.949 , and an AUC of 0.974 in multi-class detection tasks. When the number of SDN domains increases from 5 to 50, the F1-score decreases by only 1.2 percentage points. At a privacy budget of 2.0, the success rate of membership inference attacks is significantly lower than that of models without differential privacy. Attention heatmaps further show that the model can dynamically focus on key SDN features according to attack type and reveal the spatiotemporal response process of anomalies in multi-domain industrial networks. Overall, the proposed method provides a practical deployable solution for privacy-preserving intrusion detection in scalable distributed industrial information systems. However, several limitations remain, and future work is organized below in order of priority. First, and nearest-term, adaptive dynamic privacy-budget allocation should be optimized under simulation, since the fixed setting of $\varepsilon = 2.0$ is a compromise identified in Table 5 and strong differential-privacy settings can still reduce detection performance. Second, building on that foundation, defenses against active malicious-client poisoning attacks should be strengthened, since this paper mainly considers honest-but-curious servers and passive privacy attacks. Third, once these algorithmic components are strengthened, the framework should be deployed and verified on real industrial SDN field networks, addressing the gap between Mininet simulation and production networks in background traffic complexity, hardware performance, and security-policy diversity, and extending the approach to broader cloud-edge collaborative networks, 5G network slicing, and general-purpose enterprise SDN deployments.

References

[1] M. S. Ataa, E. E. Sanad, and R. A. El-khoribi, (2024) *"Intrusion Detection in Software Defined Network Using Deep Learning Approaches"* **Scientific Reports *14***: 29159. DOI: 10.1038/s41598-024-79001-1.

[2] A. H. Janabi, T. Kanakis, and M. Johnson, (2024) *"Survey: Intrusion Detection System in Software-Defined Networking"* **IEEE Access *12***: 164097–164120. DOI: 10.1109/ACCESS.2024.3493384.

[3] M. Raza, M. J. Saeed, M. B. Riaz, and M. A. Sattar, (2024) *"Federated Learning for Privacy-Preserving Intrusion Detection in Software-Defined Networks"* **IEEE Access *12***: 69551–69567. DOI: 10.1109/ACCESS.2024.3395997.

[4] H. Babbar and S. Rani, (2024) *"FRHIDS: Federated Learning Recommender Hybrid Intrusion Detection System Model in Software-Defined Networking for Consumer Devices"* **IEEE Transactions on Consumer Electronics *70***(1): 2492–2499. DOI: 10.1109/TCE.2023.3329151.

[5] R. Ben Said, Z. Sabir, and I. Askerzade, (2023) *"CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection"* **IEEE Access *11***: 138732–138747. DOI: 10.1109/ACCESS.2023.3340142.

[6] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, (2022) *"Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions"* **Computer Communications *195***: 346–361. DOI: 10.1016/j.comcom.2022.09.012.

[7] H. Alshahrani, A. Khan, M. Rizwan, M. S. Al Reshan, A. Sulaiman, and A. Shaikh, (2023) *"Intrusion Detection Framework for Industrial Internet of Things Using Software Defined Network"* **Sustainability *15***(11): 9001. DOI: 10.3390/su15119001.

[8] A. H. Janabi, T. Kanakis, and M. Johnson, (2022) *"Overhead Reduction Technique for Software-Defined Network Based Intrusion Detection Systems"* **IEEE Access *10***: 66481–66491. DOI: 10.1109/ACCESS.2022.3184722.

[9] H. Bour, M. Abolhasan, S. Jafarizadeh, J. Lipman, and I. Makhdoom, (2022) *"A Multi-Layered Intrusion Detection System for Software Defined Networking"* **Computers and Electrical Engineering *101***: 108042. DOI: 10.1016/j.compeleceng.2022.108042.

[10] A. Bhardwaj, R. Tyagi, N. Sharma, A. Khare, M. S. Punia, and V. K. Garg, (2022) *"Network Intrusion Detection in Software Defined Networking with Self-Organized Constraint-Based Intelligent Learning Framework"* **Measurement: Sensors *24***: 100580. DOI: 10.1016/j.measen.2022.100580.

[11] B. Hu, Y. Bi, M. Zhi, K. Zhang, F. Yan, and Q. Zhang, (2022) *"A Deep One-Class Intrusion Detection Scheme in Software-Defined Industrial Networks"* **IEEE Transactions on Industrial Informatics *18***(6): 4286–4296. DOI: 10.1109/TII.2021.3133300.

[12] P. Ruzafa-Alcazar, P. Fernández Saura, E. Mármol Campos, A. González-Vidal, J. L. Hernández-Ramos, J. Bernal Bernabé, and A. F. Skarmeta, (2023) *"Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT"* **IEEE Transactions on Industrial Informatics *19***(2): 1145–1154. DOI: 10.1109/TII.2021.3126728.

[13] A. Vyas, P.-C. Lin, R.-H. Hwang, and M. Tripathi, (2024) *"Privacy-Preserving Federated Learning for Intrusion Detection in IoT Environments: A Survey"* **IEEE Access *12***: 127018–127050. DOI: 10.1109/ACCESS.2024.3454211.

[14] M. S. ElSayed, N.-A. Le-Khac, M. A. Albahar, and A. Jurcut, (2021) *"A Novel Hybrid Model for Intrusion Detection Systems in SDNs Based on CNN and a New Regularization Technique"* **Journal of Network and Computer Applications *191***: 103160. DOI: 10.1016/j.jnca.2021.103160.

[15] Z. Wu, H. Zhang, P. Wang, and Z. Sun, (2022) *"RTIDS: A Robust Transformer-Based Approach for Intrusion Detection System"* **IEEE Access *10***: 64375–64387. DOI: 10.1109/ACCESS.2022.3182333.

[16] J. Chen, Y. Zhao, Q. Li, X. Feng, and K. Xu, (2023) *"FedDef: Defense Against Gradient Leakage in Federated Learning-Based Network Intrusion Detection Systems"* **IEEE Transactions on Information Forensics and Security *18***: 4561–4576. DOI: 10.1109/TIFS.2023.3297369.

[17] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, (2023) *"Deep Learning Approach for SDN-Enabled Intrusion Detection System in IoT Networks"* **Information *14***(1): 41. DOI: 10.3390/info14010041.

[18] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, (2023) *"A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE"* **IEEE Access *11***: 37131–37148. DOI: 10.1109/ACCESS.2023.3266979.

[19] E. Mármol Campos, P. Fernández Saura, A. González-Vidal, J. L. Hernández-Ramos, J. Bernal Bernabé, G. Baldini, and A. Skarmeta, (2022) *"Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges"* **Computer Networks *203***: 108661. DOI: 10.1016/j.comnet.2021.108661.

[20] S. Zavrak and M. Iskefiyeli, (2023) *"Flow-Based Intrusion Detection on Software-Defined Networks: A Multivariate Time Series Anomaly Detection Approach"* **Neural Computing and Applications *35***(16): 12175–12193. DOI: 10.1007/s00521-023-08376-5.

[21] P. Hadem, D. K. Saikia, and S. Moulik, (2021) *"An SDN-Based Intrusion Detection System Using SVM with Selective Logging for IP Traceback"* **Computer Networks *191***: 108015. DOI: 10.1016/j.comnet.2021.108015.

[22] Y. Al-Dunainawi, B. R. Al-Kaseem, and H. S. Al-Raweshidy, (2023) *"Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment"* **IEEE Access *11***: 106733–106748. DOI: 10.1109/ACCESS.2023.3319214.

[23] T. V. Phan and T. Bauschert, (2022) *"DeepAir: Deep Reinforcement Learning for Adaptive Intrusion Response in Software-Defined Networks"* **IEEE Transactions on Network and Service Management *19***(3): 2207–2218. DOI: 10.1109/TNSM.2022.3158468.

[24] T. Naqash, S. H. Shah, and M. N. Ul Islam, (2022) *"Statistical Analysis Based Intrusion Detection System for Ultra-High-Speed Software Defined Network"* **International Journal of Parallel Programming *50***(1): 89–114. DOI: 10.1007/s10766-021-00715-0.

[25] B. Buyuktanir, S. Altinkaya, G. Karatas Baydogmus, and K. Yildiz, (2025) *"Federated Learning in Intrusion Detection: Advancements, Applications, and Future Directions"* **Cluster Computing *28***(7): 473. DOI: 10.1007/s10586-025-05325-w.