FedCAD: Federated Contrastive-Aware Decoupling for Heterogeneous Power System API Security Detection

Xuhua Ai, Yiting Huang, Qi Meng, Zhaoli Chen, Yun Dong, Yuan Yin*
Digital Operation Center, Guangxi Power Grid Co., Ltd., Nanning 530023, China
*Corresponding author: Yuan Yin (e-mail: 202211088586@mail.scut.edu.cn).

ABSTRACT Digital power grids increasingly rely on API-based interaction among dispatch automation systems, distribution management systems, advanced metering infrastructure, marketing platforms, energy management systems, distributed energy resource access platforms, and edge IoT gateways. These interfaces support remote monitoring, operation command delivery, load forecasting, meter data acquisition, fault isolation, and cross-domain business collaboration, but they also introduce new attack surfaces for unauthorized command invocation, abnormal parameter tampering, replay access, excessive service calling, and sensitive grid-data leakage. Building an accurate API security detection model for such scenarios is challenging because power-grid data cannot be freely centralized: dispatch logs, equipment identifiers, customer electricity information, station topology, and operation records are distributed across regional grids, voltage levels, business departments, and terminal types, and are subject to strict privacy and operational-security constraints. However, power-system API security data usually exhibits multiple forms of heterogeneity. Label distributions vary because different regions and business systems face different proportions of normal operation, abnormal metering access, dispatch-command abuse, distribution-terminal intrusion, and new-energy access anomalies. Feature distributions vary because API traffic is strongly coupled with grid operation modes, voltage levels, device types, communication protocols, seasonal load patterns, and local gateway configurations. In addition, missing timestamps, clock drift, incomplete gateway logs, noisy alarm labels, and irregular field-device communication may corrupt the observed data. These factors lead to inconsistent power API behavior representations, shifted risk decision boundaries, and local overfitting in traditional distributed learning methods. To address these challenges, this paper proposes FedCAD (Federated Contrastive-Aware Decoupling), a federated framework for heterogeneous power system API security detection. FedCAD decouples representation learning from classifier adaptation. A hierarchical contrastive learning module aligns power API behavior representations at both instance and risk-category levels, while preserving region-specific and business-specific operational characteristics. An adaptive classifier module further adjusts decision boundaries according to local power API risk distributions through logit adjustment and local fine-tuning. The proposed framework enables collaborative security modeling across dispatch, distribution, metering, marketing, and new-energy systems without sharing raw power-grid data, improving robustness and generalization under multi-source heterogeneous conditions.

INDEX TERMS Federated Learning, Power System Cybersecurity, API Security Detection, Smart Grid, Contrastive Learning, Non-IID

I. INTRODUCTION

FEDERATED Learning (FL) [1] has emerged as a promising distributed machine learning paradigm that enables multiple clients to collaboratively train a global model without sharing their local raw data. This privacy-preserving characteristic is especially important for power system API security detection, where dispatch operation logs, distribution automation interactions, metering records, customer electricity information, station-device identifiers, and cross-domain authentication traces are distributed across regional grid companies, control centers, substations, edge gateways, and business platforms.

Federated learning has also been studied in privacy-sensitive application settings and as a general collaborative machine-learning framework [2], [3], with large-scale decentralized deployments demonstrating its practical feasibility [4]. These developments motivate its use for security analytics in power systems, where operational data must

remain under the control of the participating organizations.

With the digital transformation of power grids, API interfaces have become the basic connection layer among energy management systems (EMS), distribution management systems (DMS), supervisory control and data acquisition (SCADA) platforms, advanced metering infrastructure (AMI), marketing systems, outage management systems, electric-vehicle charging platforms, and distributed energy resource (DER) access services. These APIs carry operation commands, meter-data queries, fault alarms, equipment-status synchronization, load-control requests, and new-energy dispatch information. Once abused, they may trigger unauthorized remote operation, abnormal switching-command invocation, illegal meter-data access, replayed gateway requests, or leakage of sensitive grid operation data. Therefore, collaborative API security detection across power business domains is a practical requirement for improving grid cyber resilience.

Despite its advantages, FL faces significant challenges due to statistical heterogeneity (Non-IID) among clients [5], [6]. In real-world power system API security scenarios, client data typically exhibits three primary types of heterogeneity:

(1) Label distribution shift occurs when different clients have disparate class compositions. For example, a dispatch-control center may observe more abnormal command invocation and cross-domain privilege events, while a metering center may observe more excessive query, data scraping, and abnormal authentication failures. DER access platforms may face more inverter-control and telemetry-reporting anomalies, whereas marketing platforms may face more customer-information leakage risks.

(2) Feature distribution shift arises from power-domain-specific API structures and operation behaviors. API request fields, command types, device identifiers, voltage levels, station or feeder topology, protocol gateway configurations, seasonal load curves, and business workflows vary significantly among dispatch, distribution, metering, marketing, and new-energy systems. As a result, the same risk category, such as parameter tampering or replay access, may exhibit different feature distributions across clients [7].

(3) Data corruption is caused by missing gateway logs, clock drift between master stations and field terminals, incomplete alarm association, noise in manually confirmed security labels, and packet loss in edge communication. These factors directly affect the quality of power API behavior records and may lead local models to overfit incomplete, noisy, or incorrectly labeled operation patterns.

The standard FedAvg algorithm [1] performs simple weighted averaging of client models based on dataset sizes. However, this approach implicitly assumes that data is independently and identically distributed (IID) across clients. Under heterogeneous power API security settings, this naive aggregation leads to significant performance degradation because local models may overfit region-specific API traffic, equipment-access patterns, and business-risk distributions, causing the global model to diverge from the optimal solution.

Recent works have proposed various solutions to address FL heterogeneity. FedProx [5] adds a proximal term to the local objective to constrain local models from deviating too far from the global model. SCAFFOLD [8] employs variance reduction through control variates to correct the direction of local gradients. FedSAM [9] applies sharpness-aware minimization to find flat minima that generalize better. FedGAMMA [10] introduces consistency-constrained perturbations to enhance the generalization capability of the global model. FedMix [11] performs mixed sample augmentation by sending and receiving average data. FedRDN [12] injects statistical information from all clients into enhanced samples to reduce feature distribution differences. Nevertheless, power-grid API security detection requires not only statistical generalization, but also consistency with operational semantics such as command type, voltage level, device role, region, and business process.

While these methods have achieved notable improvements, they primarily address specific types of heterogeneity in isolation. Moreover, they treat representation learning and classifier training as a coupled process, which limits the model's ability to learn generalizable power API behavior representations while adapting to local risk distributions. We argue that the fundamental issue lies in the entanglement of representation learning and decision boundary adaptation. In power systems, a request pattern that is normal for one business domain, such as high-frequency meter-data collection during scheduled settlement, may be abnormal in another domain, such as excessive dispatch-command invocation during a stable operation period. Therefore, forcing a fully shared classifier across all clients may be suboptimal, as different regions, voltage levels, and business systems may require different decision boundaries to achieve reliable detection.

To address these limitations, we propose FedCAD (Federated Contrastive-Aware Decoupling), a novel framework that decouples the learning process into two complementary stages:

(1) **Hierarchical Contrastive Representation Learning:** We learn generalizable power API behavior features through hierarchical contrastive learning that operates at both instance and class levels. At the instance level, we encourage representations of semantics-preserving augmented views from the same API behavior sample to be similar. At the class level, we align prototypes of power API risk categories, such as unauthorized command, abnormal metering access, parameter tampering, replay invocation, and sensitive data exposure, across clients. Importantly, we introduce client-aware negative sampling that preserves

region-specific and business-specific power operation characteristics while enabling cross-client alignment.

(2) **Adaptive Classifier Adaptation:** We dynamically adjust classifier decision boundaries based on local power API risk distributions. This includes logit adjustment that compensates for risk-label imbalance and local classifier fine-tuning that allows each client to specialize its decision boundary for its own API traffic, equipment access pattern, and operation risk profile.

The main contributions of this paper are:

1. We formulate heterogeneous API security detection in power systems as a privacy-preserving federated learning problem involving dispatch, distribution, metering, marketing, and DER access domains.
2. We propose a decoupled learning framework that separates representation learning from classifier adaptation, enabling better generalization across heterogeneous power-grid API clients while maintaining local adaptability.
3. We introduce hierarchical contrastive learning that aligns power API behavior representations at both instance and risk-category levels while preserving region-specific and business-specific characteristics through client-aware negative sampling.
4. We design an adaptive classifier that dynamically adjusts decision boundaries based on local power API risk distributions through logit adjustment and local fine-tuning.
5. We conduct extensive experiments demonstrating that FedCAD outperforms seven state-of-the-art baselines across multiple heterogeneous power API security settings.

II. RELATED WORK

A. HETEROGENEOUS FEDERATED LEARNING

Addressing data heterogeneity in FL has been extensively studied. From the optimization perspective, FedProx [5] introduces a proximal term $\frac{\mu}{2}\|\theta-\theta^t\|_2^2$ to the local objective, constraining local models to stay close to the global model. This regularization prevents excessive client drift but may limit the ability of local models to adapt to their specific data distributions. SCAFFOLD [8] employs variance reduction through control variates c_k and c', maintaining state information to correct for client drift. FedDyn proposes dynamic regularization that aligns local and global solutions through gradient-based updates.

From the data augmentation perspective, FedMix [11] uses MixUp to perform mixed sample augmentation by exchanging average data between clients. FedRDN [12] retrieves statistical information from all clients and randomly injects it into enhanced samples to reduce differences between local distributions. These methods focus on modifying the input data distribution rather than the learning mechanism itself.

B. CONTRASTIVE LEARNING IN FEDERATED LEARNING

Contrastive learning has shown promising results in learning representations by maximizing agreement between similar samples and minimizing agreement between dissimilar samples. In centralized learning, SimCLR [13] and MoCo [14] have demonstrated the effectiveness of contrastive pretraining. In FL, MOON [15] uses model-level contrastive learning to maximize agreement between local and global models. However, MOON [15] contrasts at the model level rather than the representation level, which may not effectively capture semantic similarities. FedProto [16] transmits class prototypes instead of full models, achieving both communication efficiency and personalization. However, these methods typically couple representation learning with classifier training, limiting their effectiveness under heterogeneous settings.

C. POWER SYSTEM API SECURITY DETECTION

Power system API security detection differs from general web API security detection because API behavior is tightly coupled with physical-grid operation semantics. A single request may represent a meter-reading query, a dispatch instruction, a feeder automation event, a charging-station settlement operation, or a DER telemetry upload. The risk of the request depends not only on common cyber features such as source address, token status, request frequency, and response code, but also on power-domain attributes such as voltage level, station or feeder identity, device role, command type, operation time window, and business-system permission relationship. Therefore, an effective model should learn generalizable security representations while respecting local operation modes and regional grid characteristics. This motivates the proposed decoupled federated design.

D. DECOUPLED LEARNING

Decoupled learning separates representation learning from classifier training. In centralized learning, this approach has been explored for long-tailed recognition [17] and domain adaptation [18]. In contrast, the decoupling approach has received less attention in FL setting and our work attempts to fill this research gap.

III. METHODOLOGY

A. PROBLEM FORMULATION

We consider a federated power API security detection system with K clients. Each client $k \in \{1, 2, \ldots, K\}$ represents a regional grid unit, control center, substation gateway, or business platform, and possesses a local dataset $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{n_k}$ drawn from a local distribution $\mathcal{P}_k$. Each sample x_i denotes a power API behavior record composed of request metadata, authentication status, operation timestamp, business-system identifier, device or station identifier, voltage level, command or query type, response status, and traffic statistics. The label y_i denotes the corresponding risk category, such as normal access, unauthorized command

invocation, abnormal metering access, parameter tampering, replay invocation, excessive service calling, or sensitive data exposure. The global objective is to find model parameters θ that minimize the weighted average of local losses:

$$\min_{\theta} \mathcal{L}(\theta) = \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}_k(\theta), \tag{1}$$

where $n = \sum_{k=1}^{K} n_k$ is the total number of samples, and $\mathcal{L}_k(\theta) = \frac{1}{n_k} \sum_{(x,y)\in\mathcal{D}_k} \ell(f_\theta(x), y)$ is the local empirical risk.

In heterogeneous power-grid FL, the local distributions $\mathcal{P}_k$ differ across clients. We consider three types of heterogeneity:

- Label shift: $P_k(y) \neq P_j(y)$ for $k \neq j$, reflecting different risk proportions among dispatch, distribution, metering, marketing, and DER access clients.
- Feature shift: $P_k(x \mid y) \neq P_j(x \mid y)$ for $k \neq j$, reflecting different voltage levels, device types, API schemas, gateway configurations, operation schedules, and business workflows.
- Data corruption: $\widetilde{x} = x + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ or categorical fields are missing/masked, reflecting incomplete logs, clock drift, noisy alarms, and packet loss in edge communication.

B. OVERVIEW OF FEDCAD

FedCAD decouples the model f_θ into an encoder $f_\phi : \mathcal{X} \to \mathcal{Z}$ (parameterized by ϕ) that maps inputs to a representation space $\mathcal{Z} \subseteq \mathbb{R}^d$, and a classifier $g_\psi : \mathcal{Z} \to \mathbb{R}^C$ (parameterized by ψ) that maps representations to logits for C classes. The overall framework consists of two key components:

Hierarchical Contrastive Learning (HCL): Learns generalizable power API representations by contrasting samples at both instance and risk-category levels across clients. The encoder is trained to produce representations that are invariant to semantics-preserving log augmentation and consistent across clients for the same security meaning.

Adaptive Classifier Adaptation (ACA): Dynamically adjusts classifier decision boundaries based on local power API risk distributions. The classifier is adapted to each client's specific business domain, operation mode, and risk-label distribution through logit adjustment and local fine-tuning.

C. HIERARCHICAL CONTRASTIVE LEARNING

1) Instance-Level Contrastive Learning

For each power API behavior sample x from client k, we generate two augmented views $x^1 = t_1(x)$ and $x^2 = t_2(x)$ using semantics-preserving API-log augmentation $t_1, t_2 \sim \mathcal{T}$. In power-system scenarios, $\mathcal{T}$ may include timestamp jitter within an allowed operation window, noncritical field masking, feature dropout for optional gateway fields, normalization perturbation for continuous traffic statistics, and sequence cropping for repeated meter-reading or telemetry-upload requests. These augmentations preserve the risk semantics of the original API event while improving robustness to incomplete logs and local collection differences.

The encoded representations are $z^1 = f_\phi(x^1)$ and $z^2 = f_\phi(x^2)$.

The instance-level contrastive loss encourages the two views of the same sample to have similar representations while pushing away representations of different samples:

$$\mathcal{L}_{\text{inst}} = -\log \frac{\exp\big(\text{sim}(z^1, z^2)/\tau\big)}{\sum_{j\in\mathcal{N}_i} \exp(\text{sim}(z^1, z_j)/\tau)}, \tag{2}$$

where $\text{sim}(u, v) = u^{\mathrm{T}} v/(\|u\|\|v\|)$ is the cosine similarity, τ is the temperature parameter that controls the concentration of the distribution, and $\mathcal{N}_i$ is the set of negative samples, which will be discussed later.

2) Class-Level Contrastive Learning

To align representations of the same power API risk category across clients, we introduce class-level contrastive learning. For each class $c \in \{1, \ldots, C\}$, we maintain a class prototype $p_c^k \in \mathbb{R}^d$ as the average representation of all samples belonging to class c:

$$p_c^k = \frac{1}{|\mathcal{D}_c^k|} \sum_{(x_i, y_i)\in\mathcal{D}_c^k} f_\phi(x_i), \tag{3}$$

where $\mathcal{D}_c^k = \{(x_i, y_i) \in \mathcal{D}_k : y_i = c\}$ is the set of samples with label c on client k.

The global class prototype is computed by aggregating local prototypes:

$$p_c^g = \sum_{k=1}^{K} \frac{n_c^k}{n_c} p_c^k, \tag{4}$$

where $n_c^k = |\mathcal{D}_c^k|$ and $n_c = \sum_k n_c^k$.

The class-level contrastive loss encourages the local prototype of each risk category to align with its global counterpart. For example, unauthorized dispatch-command invocation should share a consistent semantic representation across different regional control centers, even if the concrete API schema, equipment identifier format, and invocation frequency differ:

$$\mathcal{L}_{\text{class}} = -\sum_{c=1}^{C} \frac{n_c^k}{n_k} \log \left[\frac{\exp\big(\text{sim}(p_c^k, p_c^g)/\tau\big)}{\sum_{c'=1}^{C} \exp(\text{sim}(p_c^k, p_{c'}^g)/\tau)} \right] \tag{5}$$

3) Client-Aware Negative Sampling

To preserve client-specific power-operation characteristics while enabling cross-client alignment, we employ client-aware negative sampling. For a positive pair (z^1, z^2) from client k, the negative set $\mathcal{N}_i$ consists of:

- **Local negatives:** Samples from different risk categories within the same regional or business-domain client.

- **Cross-client positives:** Samples from the same risk category from other clients, such as replay access observed in different substations or metering gateways, which are treated as additional positives.
- **Cross-client negatives:** Samples from different risk categories from other clients, while avoiding false negatives caused by similar power-operation semantics.

This design ensures that:

(1) the model learns to distinguish between different power API risks within each client;

(2) representations of the same risk category are aligned across clients; and

(3) local variations caused by region, voltage level, device type, and business workflow are preserved by not forcing all clients to have identical representations.

The combined hierarchical contrastive loss is:

$$\mathcal{L}_{\mathrm{HCL}} = \mathcal{L}_{\mathrm{inst}} + \lambda \mathcal{L}_{\mathrm{class}}, \tag{6}$$

where λ balances the two components.

D. ADAPTIVE CLASSIFIER ADAPTATION

1) Label Distribution Estimation

For each client k, we estimate the local power API risk-label distribution $\Pi_k \in \mathbb{R}^C$:

$$\Pi_k[c] = \frac{|\{x_i : (x_i, c) \in \mathcal{D}_k\}|}{n_k} = \frac{n_c^k}{n_k}. \tag{7}$$

The global label distribution is computed as $\Pi_g = \sum_{k=1}^{K} \frac{n_k}{n} \Pi_k$.

2) Adaptive Logit Adjustment

To handle risk-label distribution shift among power-grid clients, we introduce adaptive logit adjustment inspired by [19]. For a sample x with logit $o = g_\psi(f_\phi(x))$, the adjusted logit is:

$$\widetilde{o}[c] = o[c] + \gamma \log \frac{\Pi_k[c]}{\Pi_g[c]}, \tag{8}$$

where γ is a hyperparameter controlling the adjustment strength. This adjustment compensates for imbalance in local power API risk distributions. For instance, if abnormal metering access is rare globally but frequent in a local metering center, the classifier can adapt its decision boundary without forcing the encoder to forget global dispatch or distribution security semantics.

The cross-entropy loss with logit adjustment is:

$$\mathcal{L}_{\mathrm{CE}}^{\mathrm{adj}} = -\log \frac{\exp(\widetilde{o}[y])}{\sum_{c'=1}^{C} \exp(\widetilde{o}[c'])}. \tag{9}$$

3) Local Classifier Fine-Tuning

After global aggregation, each client fine-tunes only the classifier on local data for E_{ft} epochs while keeping the encoder frozen:

$$\psi_k^{t+1} = \psi^t - \eta \nabla_\psi \mathcal{L}_{\mathrm{CE}}^{\mathrm{adj}}(g_\psi(f_\phi(x)), y)\,. \tag{10}$$

This allows each client to specialize its decision boundary to its local power business and risk distribution without affecting the shared representation.

E. FEDCAD ALGORITHM

The complete FedCAD procedure is presented in Algorithm 1.

ALGORITHM 1.
FEDCAD FOR HETEROGENEOUS POWER SYSTEM API SECURITY DETECTION

Input K power-grid clients; local datasets $\{\mathcal{D}_k\}_{k=1}^K$; learning rates η_1 and η_2; local epochs E; fine-tuning epochs E_{ft}; rounds T; hyperparameters λ, τ, and γ.

Initialize global encoder ϕ^0 and classifier ψ^0.
Initialize global class prototypes $\{p_c^g\}_{c=1}^C$.
For each round $t = 0, 1, \ldots, T-1$:
Server broadcasts ϕ^t and $\{p_c^g\}$ to all clients.
For each client $k \in [K]$ in parallel:
Set $\phi_k \leftarrow \phi^t$ and $\psi_k \leftarrow \psi^t$; compute local risk distribution π_k.
Hierarchical Contrastive Learning: for E epochs and each batch $(x, y) \in \mathcal{D}_k$, generate augmented views x^1 and x^2, compute z^1 and z^2, compute $\mathcal{L}_{\mathrm{inst}}$ using (2) and $\mathcal{L}_{\mathrm{class}}$ using (5), set $\mathcal{L}_{\mathrm{HCL}} = \mathcal{L}_{\mathrm{inst}} + \lambda \mathcal{L}_{\mathrm{class}}$, and update ϕ_k with learning rate η_1.
Update local prototypes $\{p_c^k\}$ and send them to the server.
Adaptive Classifier Adaptation: for E_{ft} epochs and each batch $(x, y) \in \mathcal{D}_k$, compute logit o, apply (8), compute $\mathcal{L}_{\mathrm{CE}}^{\mathrm{adj}}$, and update ψ_k with learning rate η_2.
Upload ϕ_k^{t+1} and ψ_k^{t+1} to the server.
Aggregate encoder $\phi^{t+1} = \sum_k (n_k/n) \phi_k^{t+1}$ and classifier $\psi^{t+1} = \sum_k (n_k/n) \psi_k^{t+1}$.
Aggregate prototypes $p_c^g = \sum_k (n_c^k / n_c) p_c^k$.

Output Global power API security detection model ϕ^T and ψ^T.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) Datasets

We evaluate FedCAD on heterogeneous power system API security scenarios constructed from anonymized API access logs and operation-context records. Each sample contains cyber features and power-domain features, including request method, URL template, token status, source system, operation timestamp, response code, invocation interval, station or feeder identifier, device role, voltage level, business-domain tag, command or query type, and alarm association. The risk labels include normal access, unauthorized command invocation, abnormal metering access, parameter tampering, replay invocation, excessive service calling, and sensitive grid-data exposure.

2) Risk Label Distribution Shift Datasets:

- **PowerAPI-Dispatch:** API behavior records from dispatch automation and energy management systems. We use Dirichlet partitioning with $\alpha \in \{0.1, 0.6\}$ and shard-based partitioning with $s \in \{2, 5\}$ to simulate imbalanced risk distributions among regional control centers.
- **PowerAPI-Metering:** API behavior records from AMI, marketing, and customer-service access platforms. We use $\alpha \in \{0.1, 0.6\}$ and $s \in \{20, 50\}$ to simulate different proportions of normal meter reading, abnormal query, authentication failure, and sensitive-data access events across clients.

3) Feature Distribution Shift Datasets:

- **PowerAPI-Business:** API records from five business domains: dispatch, distribution automation, metering, marketing, and DER access. The same risk category may appear with different URL schemas, command types, permission structures, and operation schedules.
- **PowerAPI-Terminal:** API records from different terminal and gateway types, including master-station gateways, substation gateways, feeder terminals, charging-station platforms, and DER edge gateways. This setting evaluates robustness under device-role and communication-environment feature shifts.

4) Multiple Heterogeneous Conditions Dataset:

- **PowerAPI-LFC:** A combined power API scenario with label shift, business-domain feature shift, and log corruption. Label distribution shifts follow Dirichlet distribution ($\alpha = 0.1$), and data corruption is introduced by adding continuous-feature noise, timestamp jitter, and categorical-field masking with noise levels $\sigma^2 \in \{0.1, 0.5\}$.

5) Baselines

We compare FedCAD against seven competitive state-of-the-art methods:

- **FedAvg [1]:** Standard federated averaging.
- **FedDyn [20]:** Dynamic regularization for local-global alignment.
- **SCAFFOLD [8]:** Variance reduction with control variates.
- **FedSAM [9]:** Sharpness-aware minimization for generalization.
- **FedGAMMA [10]:** Consistency-constrained perturbations.
- **FedMix [11]:** Mixed sample augmentation.
- **FedRDN [12]:** Statistical information injection for feature alignment.

6) Implementation Details

Following the power API security detection protocol, we use:

- **Risk label distribution shift scenarios:** 100 clients representing regional units, control-center partitions, or business-system nodes, with 10% participants uniformly sampled per communication round for 1000 rounds and 5 local epochs each. We employ a lightweight API-log encoder composed of categorical embeddings, numerical feature normalization, temporal-position encoding, and multilayer perceptron blocks.
- **Feature and multiple distribution shift scenarios:** 100% client participation for 200 rounds and 5 epochs each. We adopt a hybrid sequence encoder with categorical embeddings and a temporal convolution/Transformer block to capture repeated meter-reading, telemetry-upload, and command-invocation patterns.
- **Optimizer:** SGD with learning rate 0.01, momentum 0.9, weight decay 0.0001.
- **Batch size:** 50 for local training, 128 for testing.

For FedCAD, we set $\lambda = 0.5$, $\tau = 0.5$, $\gamma = 0.1$, $E_{\text{ft}} = 2$. Data augmentation includes timestamp jitter within legal operation windows, optional-field masking, feature dropout for noncritical gateway fields, continuous traffic-statistic perturbation, and sequence cropping for repeated telemetry or metering requests.

B. MAIN RESULTS

1) Risk Label Distribution Shift

Table I presents the test accuracy comparison on PowerAPI-Dispatch and PowerAPI-Metering under risk label distribution shift. FedCAD consistently outperforms all baselines across all settings.

TABLE I:
TEST ACCURACY (%) ON POWER API RISK LABEL DISTRIBUTION SHIFT DATASETS

	PowerAPI-Dispatch				PowerAPI-Metering			
Method	$\alpha = 0.1$	$\alpha = 0.6$	$s = 2$	$s = 5$	$\alpha = 0.1$	$\alpha = 0.6$	$s = 20$	$s = 50$
FedAvg	73.82	79.73	71.05	77.29	37.87	38.98	40.22	40.35
FedDyn	76.21	82.95	62.67	81.07	40.75	44.21	42.24	43.76
SCAFFOLD	77.86	82.47	72.64	81.60	45.54	48.33	47.56	49.50
FedSAM	74.61	79.94	71.48	77.74	39.07	39.61	40.57	41.46
FedGAMMA	77.97	82.91	72.77	81.86	46.77	49.08	48.18	50.26
FedMix	72.23	74.26	67.56	75.84	34.53	36.41	35.75	37.40
FedRDN	73.56	80.64	70.45	77.04	37.91	39.38	38.95	39.82
FedCAD	**79.38**	**83.55**	**73.48**	**82.30**	**47.33**	**50.52**	**49.72**	**51.47**

Key observations:

(1) FedCAD achieves the best performance under either extreme risk-label imbalance ($\alpha = 0.1$, $s = 2$ or 20) or moderate imbalance ($\alpha = 0.6$, $s = 5$ or 50).

(2) FedGAMMA and SCAFFOLD are the strongest baselines, but FedCAD still outperforms them by effectively decoupling power API representation learning from local risk classifier adaptation.

2) Feature Distribution Shift

Table II shows results on PowerAPI-Business and PowerAPI-Terminal.

TABLE II:
TEST ACCURACY (%) ON POWER API FEATURE DISTRIBUTION SHIFT DATASETS

	PowerAPI-Business		PowerAPI-Terminal	
Method	**AVG↑**	**STD↓**	**AVG↑**	**STD↓**
FedAvg	45.30	10.01	63.40	18.61
FedDyn	45.40	9.91	63.75	19.13
SCAFFOLD	45.58	9.52	65.72	18.34
FedSAM	47.07	9.03	66.14	18.62
FedGAMMA	47.93	8.82	68.90	18.36
FedMix	45.81	9.42	65.71	19.05
FedRDN	47.50	8.90	70.44	18.32
FedCAD	**49.01**	**8.72**	**71.23**	**18.15**

FedCAD achieves the highest average accuracy and lowest standard deviation, indicating both better performance and improved fairness across power business domains and terminal types. The hierarchical contrastive learning effectively aligns representations across different feature distributions while preserving local operation characteristics.

3) Multiple Heterogeneous Conditions

Table III presents results on PowerAPI-LFC with simultaneous risk-label shift, feature shift, and log corruption.

TABLE III:
TEST ACCURACY (%) ON POWERAPI-LFC WITH MULTIPLE HETEROGENEITIES

	PowerAPI-LFC (0.1)		PowerAPI-LFC (0.5)	
Method	**AVG↑**	**STD↓**	**AVG↑**	**STD↓**
FedAvg	56.53	17.72	48.62	19.75
FedDyn	57.98	17.20	50.28	19.46
SCAFFOLD	59.15	16.13	50.97	18.55
FedSAM	59.31	16.06	50.82	19.32
FedGAMMA	61.83	15.63	53.39	18.50
FedMix	59.45	15.74	50.98	19.58
FedRDN	60.51	15.50	52.16	18.74
FedCAD	**62.50**	**15.25**	**54.04**	**18.41**

FedCAD demonstrates particularly strong performance under multiple heterogeneous power API conditions, achieving 0.65–0.67% improvement over the best baseline. This validates the effectiveness of the decoupled approach in handling complex real-world power-grid security scenarios.

C. ABLATION STUDY

We conduct comprehensive ablation studies on PowerAPI-Dispatch ($\alpha = 0.1$) to validate each component's contribution. Results are shown in Table IV.

TABLE IV:
ABLATION STUDY ON POWERAPI-DISPATCH ($\alpha = 0.1$)
CA: INSTANCE-LEVEL CL; LA: LOGIT ADJUSTMENT; HA: CLASS-LEVEL CL; HC: CLIENT-AWARE NEGATIVE SAMPLING

Configuration	**Accuracy (%)**
FedCAD (Full)	**79.38**
w/o CA (instance-level CL)	76.82 (-2.56)
w/o LA (logit adjustment)	77.45 (-1.93)
w/o HA (class-level CL)	78.18 (-1.20)
w/o HC (client-aware sampling)	77.92 (-1.46)
w/o local fine-tuning	78.89 (-0.49)

Key findings:

(1) Instance-level contrastive learning (CA) contributes the most (2.56%), demonstrating the importance of learning invariant power API behavior representations.

(2) Logit adjustment (LA) provides 1.93% improvement, validating its effectiveness in handling local risk-label imbalance.

(3) Class-level alignment (HA) contributes 1.20%, showing the benefit of cross-client power-risk semantic alignment.

(4) Client-aware negative sampling (HC) provides 1.46% improvement, indicating its importance in preserving region-specific and business-specific power operation characteristics.

D. PARAMETER SENSITIVITY ANALYSIS

We analyze the sensitivity of key hyperparameters on PowerAPI-Dispatch ($\alpha = 0.1$). The results are shown in Fig. 1.

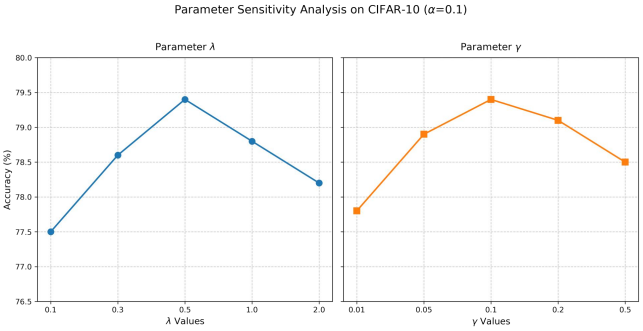

FIGURE 1.
Parameter sensitivity analysis on PowerAPI-Dispatch ($\alpha = 0.1$). The model accuracy is shown for different values of (a) λ and (b) γ.

Effect of λ (instance-level CL vs. class-level CL): Performance increases with λ up to 0.5, then plateaus. Setting $\lambda = 0.5$ provides the best balance between instance discrimination and class alignment. Higher values may overemphasize class alignment at the expense of instance discrimination.

Effect of γ (logit adjustment strength): Performance is relatively stable for $\gamma \in [0.05, 0.2]$, with optimal value at $\gamma = 0.1$. Too small values provide insufficient adjustment, while too large values may overcompensate.

V. DISCUSSION

The experimental results show that FedCAD is effective across label-distribution shift, feature-distribution shift, and the combined PowerAPI-LFC setting. The largest and most consistent gains occur when heterogeneous conditions coexist, suggesting that representation alignment and local decision-boundary adaptation address complementary aspects of power API security detection. In particular, the hierarchical contrastive component improves the transferability of security representations across dispatch, metering, business, and terminal domains, whereas adaptive logit adjustment and local fine-tuning help each client accommodate its own risk prevalence and operational context.

The ablation results further clarify this complementarity. Removing instance-level contrastive adaptation causes the largest accuracy reduction, indicating that robust instance

representations are the foundation for collaborative detection. The decreases observed after removing logit adjustment, hierarchical alignment, or client-aware sampling show that neither global alignment nor local adaptation alone is sufficient. This is especially relevant in power systems, where identical API actions can carry different security implications depending on the device role, voltage level, business process, and time window. A fully shared classifier may therefore be less suitable than a shared encoder combined with locally adaptable decision boundaries.

Several limitations should be considered when interpreting these results. First, the experiments use anonymized, constructed power API scenarios; future work should validate the framework using broader operational datasets from additional grid regions and longer observation periods. Second, the present evaluation focuses on detection accuracy and cross-domain stability. Deployment studies should additionally measure communication cost, training latency, concept drift, and resilience to malicious or unreliable clients. Third, although raw records are not exchanged, model updates and prototypes may still reveal information; incorporating secure aggregation, differential privacy, or robust aggregation would further strengthen the practical privacy and security guarantees. These directions would help translate FedCAD from an experimental framework into a dependable component of production power-grid security operations.

VI. CONCLUSION

We proposed FedCAD, a federated learning framework for heterogeneous power system API security detection. By separating hierarchical contrastive representation learning from adaptive classifier adaptation, FedCAD can collaboratively learn from dispatch, distribution, metering, marketing, and DER access clients without sharing raw power-grid data. The framework effectively handles risk-label distribution shift, power-domain feature shift, and log corruption caused by incomplete gateway records or edge communication noise. Extensive experiments demonstrate state-of-the-art performance across heterogeneous power API security settings, with particularly strong results under multiple simultaneous heterogeneity conditions. These results indicate that representation-classifier decoupling is a practical direction for privacy-preserving, cross-region, and cross-business-domain security detection in digital power grids.

FUNDING

This work was supported by the Major Science and Technology Project GXKJXM20240043: ‘Research and Demonstration of API Security Governance Framework and Intelligent Protection Key Technologies for Application Systems.’

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2017, pp. 1273–1282.

[2] N. Rieke, J. Hancox, W. Li, et al., “The future of digital health with federated learning,” *NPJ Digital Medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[4] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, et al., “Federated learning for mobile keyboard prediction,” *arXiv:1811.03604*, 2018.

[5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” in *MLSys*, 2020, pp. 429–450.

[6] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-IID data,” *arXiv:1806.00582*, 2018.

[7] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, “FedBN: Federated learning on non-IID features via local batch normalization,” in *ICLR*, 2021.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, “SCAFFOLD: Stochastic controlled averaging for federated learning,” in *ICML*, 2020, pp. 5132–5143.

[9] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu, “Generalized federated learning via sharpness aware minimization,” in *ICML*, 2022, pp. 18250–18280.

[10] R. Dai, X. Yang, Y. Sun, L. Shen, X. Tian, M. Wang, and Y. Zhang, “FedGAMMA: Federated learning with global sharpness-aware minimization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 12, pp. 17479–17492, 2024.

[11] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, “FedMix: Approximation of mixup under mean augmented federated learning,” in *ICLR*, 2021.

[12] Y. Yan, H. Fu, Y. Li, et al., “A simple data augmentation for feature distribution skewed federated learning,” in *CVPR*, 2025, pp. 25749–25758.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *ICML*, 2020, pp. 1597–1607.

[14] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, 2020, pp. 9729–9738.

[15] Q. Li, B. He, and D. Song, “Model-contrastive federated learning,” in *CVPR*, 2021, pp. 10713–10722.

[16] Y. Tan, G. Long, J. Ma, L. Liu, T. Zhou, and J. Jiang, “Federated learning from pre-trained models: A contrastive learning approach,” in *NeurIPS*, 2022, pp. 19332–19344.

[17] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, “Decoupling representation and classifier for long-tailed recognition,” in *ICLR*, 2019.

[18] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, “Adaptive batch normalization for practical domain adaptation,” *Pattern Recognition*, vol. 80, pp. 109–117, 2018.

[19] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, “Long-tail learning via logit adjustment,” in *ICLR*, 2020.

[20] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, “Federated learning based on dynamic regularization,” in *ICLR*, 2021.