Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082704

ARTICLE

Cross-Domain Robust Dynamic Trust Evaluation for Industrial Internet of Things Edge Nodes

Qiuguo Guan and Zhiyu Ren*

School of Cryptography Engineering, Engineering University, Zhengzhou, China

*Corresponding Author: Zhiyu Ren. Email: ren_ktzy@163.com

Received: 23 March 2026; Accepted: 19 May 2026; Published: 15 June 2026

ABSTRACT: To address trust-score drift and unsafe online adaptation under cross-domain attack-contaminated streams in Industrial Internet of Things (IIoT) edge environments, this paper proposes a risk-aware lightweight test-time adaptation (TTA) framework, named RaL-TTA, for dynamic trust evaluation of edge nodes. RaL-TTA constructs a low-dimensional robust feature space and a source-domain normal-entropy reference baseline, and performs selective online maintenance in the target domain through Kolmogorov–Smirnov (KS) drift detection, SafeBrake risk gating, Adaptive Batch Normalization (AdaBN) anchor protection, and budgeted sample-level safeguards. Low-risk batches are adapted by updating only lightweight Batch Normalization (BN) parameters, whereas high-risk batches freeze online updates and invoke anchor-based protective inference. Experiments on Edge-IIoTset show that RaL-TTA substantially improves perturbation-stage attack detection and false-positive control compared with general TTA baselines while maintaining post-perturbation stability. In the main Edge-IIoTset setting, RaL-TTA achieves a perturbation-stage true positive rate (TPR) of 1.0000, false positive rate (FPR) of 0.0410, F1-score of 0.9544, and accuracy of 0.9713, while updating only 192 online parameters. External validation on X-IIoTID,a connectivity- and device-agnostic intrusion dataset for IIoT, further evaluates cross-service generalization under Modbus, Message Queuing Telemetry Transport (MQTT), and WebSocket target services. Additional sensitivity, startup-window robustness, calibration, and runtime-overhead analyses further characterize the stability, deployment assumptions, trust-score reliability, and edge-side feasibility of the proposed framework.

KEYWORDS: Industrial Internet of Things; dynamic trust evaluation; cross-domain intrusion detection; test-time adaptation; risk-constrained adaptation; edge security

1 Introduction

The Industrial Internet of Things (IIoT) has developed rapidly with the accelerated integration of next-generation information technology and industrial manufacturing. It extends computing capabilities from cloud infrastructures to edge devices and has become an important infrastructure for industrial intelligence. Its security and reliability are directly related to the continuous and stable operation of physical production processes [1–3]. With the ubiquitous deployment of heterogeneous sensors and actuators, IIoT systems face complex security challenges arising from the coexistence of open connectivity and closed-loop physical processes. Dynamic trust evaluation is an important way to complement traditional static defense in heterogeneous edge scenarios and to support adaptive access control and edge-side risk decision-making [4,5].

Most existing studies on IIoT trust evaluation and industrial intrusion detection follow a static offline training mode [6–8]. Whether traditional machine-learning methods, such as random forests and support vector machines, or deep neural-network-based anomaly detection models are employed, their performance usually depends heavily on the assumption that the training data and deployment data follow an independent and identically distributed (IID) setting [9,10]. Such models are typically trained once in the source domain, such as a laboratory environment, and then fixed, which limits their adaptability to non-stationary environments. However, in actual industrial settings, edge nodes face severe domain-shift challenges. The heterogeneity of underlying communication protocols, such as migration between Message Queuing Telemetry Transport (MQTT) and Modbus over Transmission Control Protocol (Modbus/TCP), and the time-varying nature of production conditions can make the target-domain data distribution deviate from the source-domain prior [6,8]. Violating the IID assumption can therefore degrade the detection performance of a source-domain model in the target domain and distort the corresponding trust scores.

To alleviate the degradation of source-domain models in the target domain, researchers have regarded cross-domain trust evaluation as a distribution-transfer problem under non-stationary environments and introduced mechanisms such as domain adaptation (DA) to reduce the impact of domain shift [11,12]. For example, Adaptive Batch Normalization (AdaBN) achieves distribution alignment by re-estimating normalization statistics in the target domain [13]; Source Hypothesis Transfer (SHOT) freezes the source-domain classifier without accessing source data and iteratively optimizes the target-domain feature extractor [14]. To address continuous distribution changes in industrial edge environments, methods such as EdgeFD combine drift detection with model-weight integration to reduce the overhead caused by frequent fine-tuning and to alleviate catastrophic forgetting [11]. However, such methods often still rely on phased adaptation processes or iterative optimization and are sensitive to statistical estimates from small batches, making it difficult to meet the requirements of edge-node security detection in online unlabeled scenarios. Therefore, the research focus has gradually shifted to test-time adaptation (TTA) methods that can operate without retraining after deployment.

TTA reduces the deployment cost of adaptation because it does not require source-domain data after deployment [15]. AdaBN rapidly aligns distributions by re-estimating Batch Normalization (BN) statistics [13]; fully test-time adaptation by entropy minimization (TENT) optimizes model parameters online through entropy minimization to increase prediction confidence in the target domain [16]. For continuously changing target-domain distributions, methods such as continual test-time adaptation (CoTTA) suppress error accumulation and catastrophic forgetting through teacher-student consistency, enhanced averaging, and random recovery [12]. Other studies construct adaptation objectives from energy functions and iterative sampling [17], or monitor entropy drift online and perform entropy-distribution matching for more robust trigger-based adaptation [18]. However, directly applying general TTA methods to adversarial IIoT edge environments still faces resource and security constraints. Frequent backpropagation, multiple data augmentations, or iterative sampling can increase inference latency on resource-limited edge nodes. In addition, entropy minimization may compress prediction uncertainty and cause overconfident erroneous convergence or model contamination when malicious attacks or high-noise disturbances appear, thereby reducing the reliability of trust scores [10].

However, directly applying existing TTA methods to IIoT trust evaluation remains insufficient. AdaBN mainly recalibrates BN statistics and does not explicitly distinguish benign domain shift from attack-contaminated drift. TENT performs entropy minimization during testing, but may become overconfident on abnormal or adversarial target samples. Protected Online Entropy Matching (POEM) improves online entropy matching, but it is not specifically designed for dynamic trust evaluation under attack-risk constraints. In contrast, the proposed risk-aware lightweight test-time adaptation (RaL-TTA) framework

introduces a risk-aware maintenance strategy that combines entropy-distribution shift detection, SafeBrake risk gating, AdaBN-based anchor protection, and budgeted sample-level safeguards. Therefore, the proposed method is not only an adaptation mechanism but also a security-oriented dynamic trust evaluation framework for edge-side IIoT streams. Table 1 summarizes the key differences.

Table 1: Comparison between RaL-TTA and representative TTA methods.

Method	Primary Adaptation Target	Risk Gate	Anchor Protection	Sample Safeguard	Trust-Score Output
AdaBN	BN statistics	No	No	No	No
TENT	BN affine parameters	No	No	No	No
POEM	Entropy matching	Partial	No	No	No
POEM+SafeBrake	Entropy matching + risk gate	Yes	Partial	Partial	No
RaL-TTA	BN maintenance + protection	Yes	Yes	Yes	Yes

Note: "Partial" indicates that the corresponding mechanism is only indirectly or incompletely included and is not formulated as a complete trust-score safeguard.

To address resource constraints, cross-domain distribution shifts, and attack-stream contamination in unlabeled online updates for IIoT edge nodes, this paper proposes the RaL-TTA framework for cross-domain dynamic trust evaluation. Unlike general TTA strategies that lack explicit risk constraints, RaL-TTA builds an "offline trust baseline–online risk gating" mechanism. In the source-domain stage, it uses the low-dimensional feature set for Industrial Internet of Things (LoFT-IIoT) [19] and label-smoothed training to establish a normal-entropy reference baseline. In the target-domain stage, it combines shift detection with SafeBrake risk gating, freezes updates for high-risk batches, invokes Adaptive Batch Normalization (AdaBN) anchor protection, and performs restricted BN maintenance only for low-risk batches. Budgeted sample-level arbitration is used only as a supplementary safety boundary for a small number of highly ambiguous samples. This framework enhances detection performance, trust-score stability, and maintenance security in cross-domain online streams while controlling online overhead.

The main contributions of this work are summarized as follows:

1. We formulate security-oriented dynamic trust evaluation for IIoT edge nodes under cross-domain online streams, where the task-level trust score is derived from the estimated attack risk.
2. We develop RaL-TTA, a lightweight TTA framework that combines a low-dimensional source-domain trust baseline with risk-constrained online maintenance for edge-side deployment.
3. We introduce a protective online adaptation strategy integrating KS-based shift detection, SafeBrake risk gating, AdaBN-calibrated anchor inference, and budgeted sample-level safeguards to reduce unsafe adaptation under attack-contaminated streams.
4. We validate the proposed framework on Edge-IIoTset and an external service-holdout setting based on X-IIoTID, a connectivity- and device-agnostic intrusion dataset for IIoT, with ablation, sensitivity, calibration, startup-window robustness, and runtime-overhead analyses.

2 RaL-TTA Cross-Domain Dynamic Trust Evaluation Framework

2.1 System Architecture

To address the domain-shift problem caused by cross-protocol communication and continuous online streaming in IIoT edge nodes, this paper constructs the RaL-TTA cross-domain dynamic trust evaluation

framework, as shown in Fig. 1. This framework consists of two phases: offline trust modeling in the source domain and online risk-constrained maintenance in the target domain. In the source-domain phase, labeled traffic is taken as input, and low-dimensional feature selection is completed through LoFT-IIoT [19], followed by training a lightweight trust evaluation model under the label-smoothing constraint. Meanwhile, the empirical cumulative distribution function (ECDF) of normal entropy is constructed based only on normal samples in the source domain, serving as a reference baseline for the target-domain online phase. Before deployment, a short target-domain startup window collected during controlled initialization is used for AdaBN anchor-model calibration, target-domain normal-reference statistics estimation, and threshold initialization. In the target-domain phase, unlabeled online streams are taken as input. First, Kolmogorov–Smirnov (KS)-based entropy-shift detection is executed, and then SafeBrake determines the risk status in combination with batch volatility. For low-risk batches, only restricted maintenance of BN affine parameters is performed, while for high-risk batches, updates are frozen and anchor protection is invoked. Sample-level arbitration is used only as a supplementary safety boundary. Finally, dynamic trust scores are output at the task level.

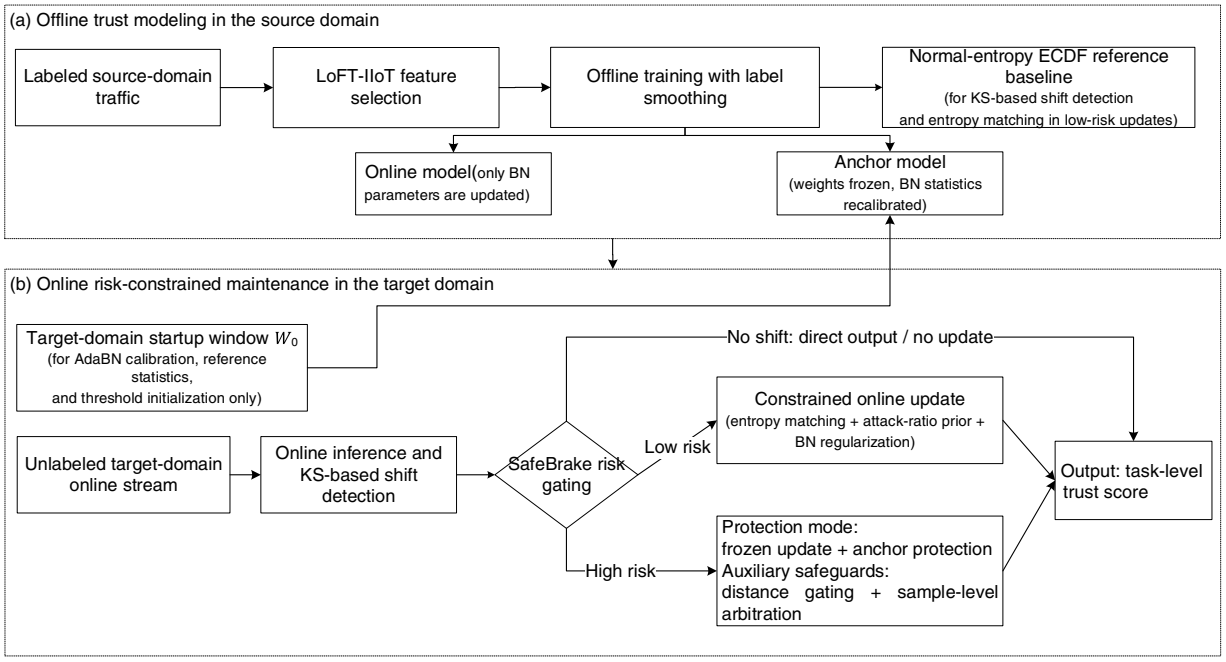

Figure 1: Overall framework of RaL-TTA for cross-domain dynamic trust evaluation. The source-domain phase constructs a low-dimensional feature space, trains the TrustMLP model, and builds a normal-entropy ECDF reference baseline. The target-domain phase processes unlabeled online batches, performs KS-based entropy-shift detection and SafeBrake risk gating, updates only BN affine parameters for low-risk batches, and invokes the AdaBN-calibrated anchor model with sample-level safeguards for high-risk batches. The startup window W_0 is used only for calibration, reference-statistics estimation, and threshold initialization.

2.2 Source-Domain Trust Baseline Construction

The construction of the source-domain trust baseline includes three steps: feature selection, model training, and estimation of the normal-entropy baseline. First, in the candidate feature space after removing protocol-specific fields, LoFT-IIoT [19] is used to select low-dimensional robust features to reduce reliance on protocol identifiers in cross-domain scenarios. Second, a lightweight multilayer perceptron (MLP) trust

evaluator with label smoothing is trained on the selected low-dimensional feature space. Finally, a normal-entropy ECDF is constructed solely from the predicted entropy of normal samples in the source domain, serving as the normal reference baseline for the target-domain online phase.

2.3 Online Risk-Aware Adaptation Mechanism

During the online phase, edge nodes receive unlabeled target-domain traffic in online batches. First, the prediction entropy of the current batch is calculated, and the KS statistic is used to measure the discrepancy between the current batch and the source-domain normal-entropy baseline. Then, SafeBrake determines the current state as no drift, low-risk drift, or high-risk drift according to both entropy-distribution shift and batch-level feature volatility.

If no significant drift is detected, the current batch is directly evaluated by the online model, and the model state remains unchanged. If a low-risk drift is detected, restricted online maintenance is executed by updating only the BN affine parameters through entropy distribution alignment, attack-ratio prior regularization, and BN regularization. If a high-risk drift is detected, the online model update is frozen, and the AdaBN-calibrated anchor model is invoked to perform protective inference. Distance gating and budget-based sample-level arbitration are used only as supplementary security boundaries for a small number of highly ambiguous samples, thereby suppressing contamination-induced erroneous adaptation.

3 Algorithm Design

3.1 Problem Formulation and Notation

In IIoT edge scenarios, edge nodes continuously receive unlabeled network traffic from the target environment. This study considers cross-domain dynamic trust evaluation in a setting where the source domain is labeled, the target domain is unlabeled, and the online distribution changes over time. The goal is to balance anomaly detection capability, output stability, and online maintenance security in continuous online streams.

Let the labeled source-domain reference set be defined as

$$\mathcal{D}_s = \mathcal{D}_H \cup \mathcal{D}_A = \{(x_i^s, y_i^s)\}_{i=1}^{N_S}, \quad y_i^s \in \{0,1\}, \tag{1}$$

where $\mathcal{D}_H = \left\{x_i^{s,0}\right\}_{i=1}^{N_H}$ denotes the set of normal samples in the source domain, and $\mathcal{D}_A = \left\{x_j^{s,1}\right\}_{j=1}^{N_A}$ denotes the set of attack samples in the source domain. The unlabeled online stream from the target domain is denoted by

$$\mathcal{S}_T = \left\{x_t^T\right\}_{t=1}^{\infty}. \tag{2}$$

To reduce the redundancy of the original traffic features and enhance the cross-domain transferability, the samples are mapped to a low-dimensional feature space through a feature mapping function $\Phi(\cdot)$:

$$z = \Phi(x), \quad z \in \mathbb{R}^m, \tag{3}$$

where m is the retained feature dimension.

A lightweight trust evaluator $\theta_{\mathrm{src}}(\cdot)$ is trained in a supervised manner on the source domain. During online deployment, the model state corresponding to the t-th batch is denoted by θ_t. For any sample x, its attack risk is defined as

$$r_t(x) = P_{\theta_t}(y = 1 \mid \Phi(x)), \tag{4}$$

where $P_{\theta_t}(y = 1 \mid \Phi(x))$ is the predicted probability that the current online model θ_t assigns sample x to the attack class.

Accordingly, the task-level trust score is defined as

$$T_t(x) = 1 - r_t(x) = P_{\theta_t}(y = 0 \mid \Phi(x)), \tag{5}$$

where $T_t(x) \in [0, 1]$, and a larger value indicates that the sample is more trustworthy under the current online model state. Here, trust assessment refers to task-level dynamic trust-score output for edge-side security decision-making, rather than a general entity-reputation propagation model. Its dynamic nature arises from the fact that the model state θ_t changes over online batches.

To characterize the uncertainty of the model prediction for the current sample, the predictive entropy is introduced:

$$h_t(x) = -\sum_{c=0}^{1} P_{\theta_t}(y = c \mid \Phi(x)) \log P_{\theta_t}(y = c \mid \Phi(x)). \tag{6}$$

Here, $P_{\theta_t}(y = c \mid \Phi(x))$ denotes the probability predicted by the current online model θ_t that sample x belongs to class c, and $h_t(x)$ denotes the predictive entropy of sample x at time t.

During the online stage, target-domain traffic arrives batch by batch. Let the current batch at time t be

$$\mathcal{B}_t = \left\{x_{t,j}^T\right\}_{j=1}^{b_t}, \tag{7}$$

where b_t is the batch size.

Based on the predictive entropy of normal source-domain samples, the empirical cumulative distribution function (ECDF) is defined as

$$F_H(h) = \frac{1}{N_H} \sum_{i=1}^{N_H} \mathbb{I}\left(h_{\theta_{\text{src}}}\left(x_i^{s,0}\right) \le h\right), \tag{8}$$

where $F_H(h)$ represents the ECDF of predictive entropy over normal source-domain samples, $h_{\theta_{\text{src}}}(x_i^{s,0})$ is the predictive entropy of the source-domain reference model θ_{src} on the normal sample $x_i^{s,0}$, and N_H is the number of normal source-domain samples.

Similarly, the empirical entropy distribution of the current batch is defined as

$$\widehat{F}_t(h) = \frac{1}{b_t} \sum_{j=1}^{b_t} \mathbb{I}\left(h_t\left(x_{t,j}^T\right) \le h\right), \tag{9}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

The distribution shift of the current batch with respect to the normal source-domain reference is then defined as

$$\Delta_t = \sup_h \left|\widehat{F}_t(h) - F_H(h)\right|. \tag{10}$$

A larger Δ_t indicates a more pronounced deviation of the current batch from the normal source-domain reference in terms of model uncertainty and can therefore serve as evidence of potential domain change during online deployment.

3.2 Construction of the Source-Domain Reference Model and Normal Baseline

3.2.1 Lightweight Feature Construction Based on LoFT-IIoT

IIoT traffic features typically exhibit significant cross-scale differences, with large variations across different fields. Such scale differences can make statistical estimation vulnerable to extreme values. Meanwhile, some application-layer fields in heterogeneous protocols have strong protocol specificity. If directly used as input, these fields may cause the model to over-rely on protocol-specific semantics, thereby weakening generalization in cross-domain scenarios. To address these issues, this paper adopts a lightweight feature construction strategy based on LoFT-IIoT [19] to filter and reduce the dimension of the original traffic features. The overall process is shown in Fig. 2.

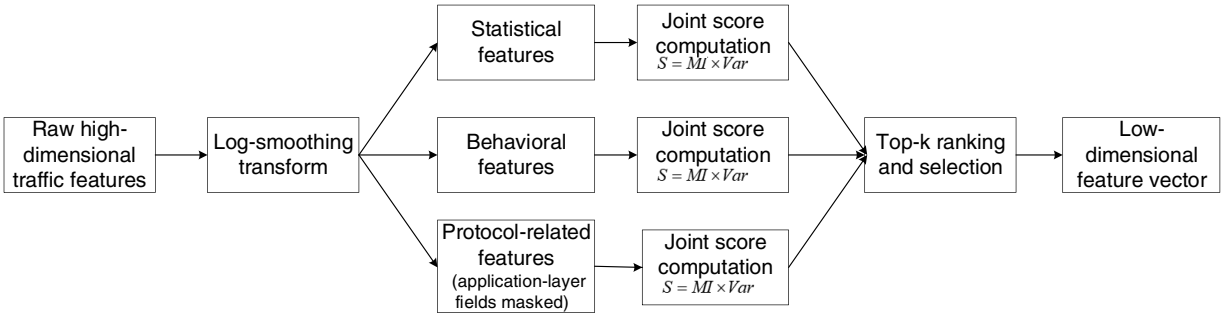

Figure 2: LoFT-IIoT-based lightweight feature construction pipeline. Raw high-dimensional traffic features are first filtered to remove labels, timestamps, host identifiers, and protocol-specific shortcut fields, and log-smoothing is applied to reduce scale differences and extreme-value effects. Candidate features are grouped into statistical, behavioral, and protocol-related categories, scored by the mutual-information–variance joint score, and ranked to select the top-k transferable features for cross-domain trust evaluation.

First, the numerical fields are retained from the original traffic features, while the label column, time column, host identification fields, and sequence-number-like fields that may introduce identity shortcuts are removed. Meanwhile, the protocol-specific fields of MQTT, Hypertext Transfer Protocol (HTTP), Modbus, Domain Name System (DNS), Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP) are masked, and only transferable low-level statistical and behavioral features are retained. Subsequently, logarithmic smoothing is performed on the candidate numerical features:

$$f_j^{\log} = \log\left(1 + \left|f_j\right|\right), \tag{11}$$

which compresses scale differences and reduces the influence of extreme values.

Second, the candidate features are classified into three semantic buckets, namely statistics, behavior, and protocol, based on the field semantics. A joint scoring strategy of mutual information and variance is adopted to describe their category discrimination ability and information activity. For the j-th candidate feature f_j, the overall score is defined as

$$s_j = I(f_j; y) \cdot \mathrm{Var}\left(f_j^{\log}\right), \tag{12}$$

where $I(f_j; y)$ denotes the mutual information between the feature and the class label, and $\mathrm{Var}(f_j^{\log})$ denotes the sample variance of the feature in the log domain.

Finally, features are pre-selected within each semantic bucket according to the joint score, and global ranking is then used for supplementation and trimming to obtain the low-dimensional feature subset $\mathcal{M}$ for subsequent model training, thereby forming the final feature mapping $\Phi(\cdot)$.

3.2.2 TrustMLP: A Lightweight Trust Assessment Model

After obtaining the low-dimensional feature representation, a lightweight MLP, named TrustMLP, is constructed as the source-domain trust evaluator. Let the model parameters be denoted by θ_{src}. The network consists of an input layer, two hidden layers, and an output layer, with width $m \to 64 \to 32 \to 2$, where m is the input dimension after feature selection, as shown in Fig. 3. Batch Normalization (BN) and Rectified Linear Unit (ReLU) activation are introduced after both hidden layers to improve training stability and to provide interfaces for the restricted BN updates used in the subsequent online stage.

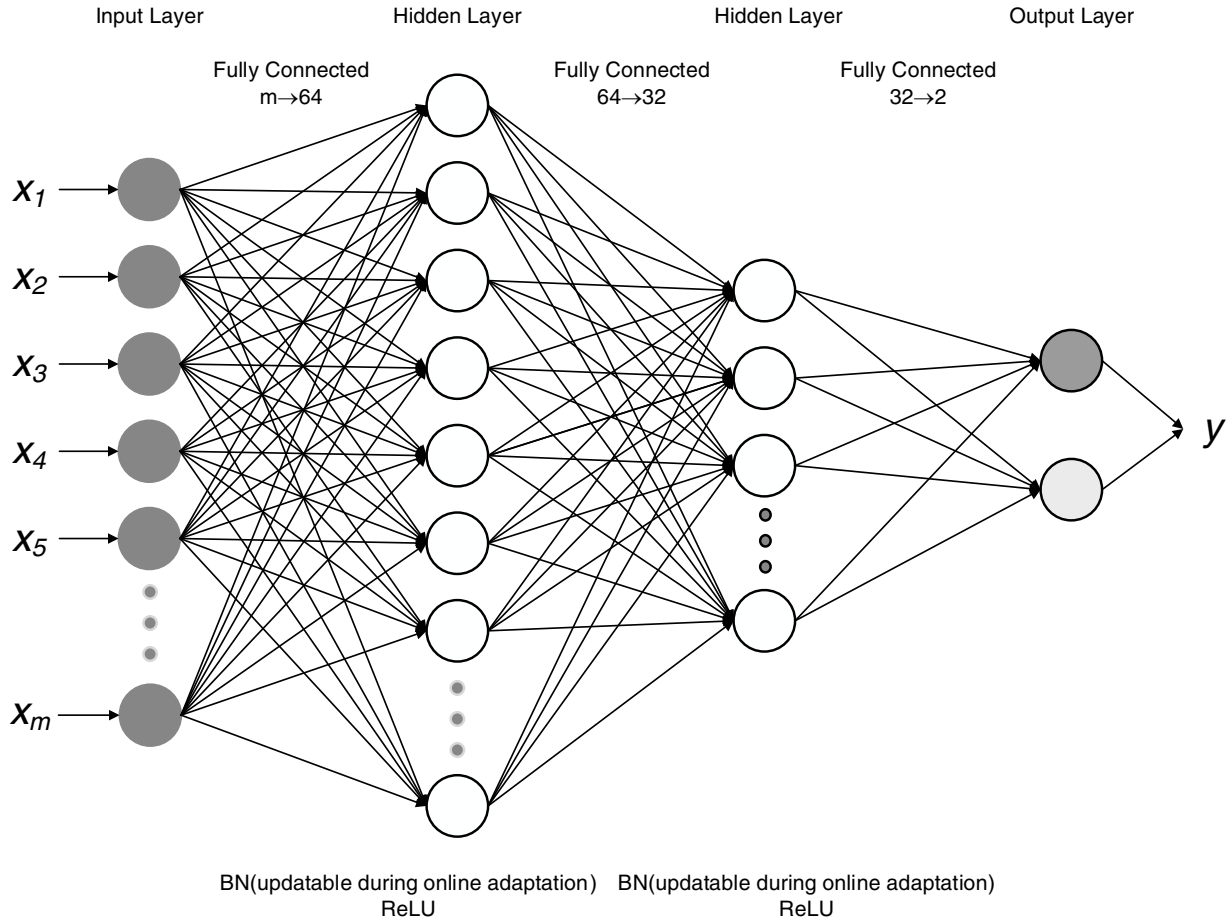

Figure 3: TrustMLP architecture for task-level trust-score estimation. The selected m-dimensional feature vector is passed through two fully connected hidden layers with Hypertext Transfer Protocol (HTTP) Rectified Linear Unit (ReLU) activation, followed by a binary output layer. The attack-class probability is used as the risk score, the normal-class probability is used as the trust score, and only BN affine parameters are updated during online adaptation.

Given the input feature $z = \Phi(x)$, the model outputs a logits vector $g_{\theta_{\text{src}}}(z) \in \mathbb{R}^2$. After the Softmax mapping, the predicted probability for class c is obtained as

$$P_{\theta_{\text{src}}}(y = c \mid z) = \frac{\exp\left(g_{\theta_{\text{src}},c}(z)\right)}{\sum_{k=0}^{1} \exp\left(g_{\theta_{\text{src}},k}(z)\right)}, \quad c \in \{0,1\}. \tag{13}$$

Here, $g_{\theta_{\text{src}},c}(z)$ denotes the logit output of the model for class c. In the binary setting considered in this work, $c = 0$ denotes normal and $c = 1$ denotes attack. Since both the attack risk and the task-level trust

score are derived from the same output probability space, no additional scoring head is required, which helps reduce inference overhead.

3.2.3 Offline Training with Label Smoothing

If the standard cross-entropy loss is directly adopted for source-domain supervised training, the model is prone to output extremely high or low probabilities close to 0 or 1 in the later stage of training, thereby causing an overconfidence problem [20]. For scenarios where a statistical baseline needs to be constructed based on the prediction entropy subsequently, this will cause the entropy values of normal samples to overly concentrate in the low range, weakening the statistical sensitivity of the entropy distribution to domain shifts. Therefore, a label smoothing strategy is introduced in the source-domain offline training stage.

Let the number of classes be C, the smoothing factor be ε, and the ground-truth class be y. The smoothed target value for the k-th class is defined as

$$y_k^{\mathrm{LS}} = \begin{cases} 1-\varepsilon, & k = y, \\ \dfrac{\varepsilon}{C-1}, & k \neq y. \end{cases} \tag{14}$$

Here, y_k^{LS} denotes the k-th component of the smoothed label vector, and k is the class index.

Based on this, offline training is completed by minimizing the cross-entropy between the predicted distribution and the soft label distribution. Before the features are input into the model, logarithmic compression and standardization processing are still adopted to eliminate the scale differences among different features. After the offline training is completed, the model parameters with the best performance on the validation set are selected as the source-domain reference model θ_{src}.

3.2.4 Construction of the Source-Domain Normal-Entropy Baseline

To build the normal reference baseline required for online shift detection in the target domain, predictive entropy is computed only over the normal source-domain sample set $\mathcal{D}_H$ after the source-domain reference model has been obtained. The resulting normal-entropy set is

$$\mathcal{E}_H = \left\{ h_{\theta_{\mathrm{src}}}\left(x_i^{s,0}\right) \right\}_{i=1}^{N_H}. \tag{15}$$

Based on the entropy set in Eq. (15), the corresponding empirical distribution function is still defined by Eq. (8).

This distribution characterizes the typical uncertainty structure of the model under normal behavior conditions and can be used as a reference to determine the degree of distribution shift in the target domain during the online phase. It is necessary to emphasize that the entropy baseline constructed in this paper is only derived from normal samples in the source domain and does not include attack samples, in order to reduce the contamination of abnormal samples on the normal reference distribution.

3.3 Risk-Constrained Online Trust Maintenance Mechanism

3.3.1 Calibration with a Target-Domain Normal Window

To enhance the statistical matching in the early stage of deployment, this paper introduces an explicit deployment assumption: a short controlled trial period containing normal target-domain traffic is available

at the beginning of deployment. The corresponding target-domain normal calibration window is denoted by

$$W_0 = \left\{x_i^{T,0}\right\}_{i=1}^{N_0}, \tag{16}$$

where W_0 is not used for supervised training. Instead, it is used only for target-domain statistical calibration and reference-baseline estimation, and it does not overlap with the online evaluation stream.

In practical IIoT deployment, such a startup window can be collected during system commissioning, scheduled maintenance, device restart, or a short trusted initialization period in which the production process operates under known normal conditions. This assumption does not require attack labels and does not expose target-domain class labels to training or online updates. If a clean startup window cannot be guaranteed, the system should operate in a conservative mode by disabling online updates until operator-confirmed normal traffic is available. The startup-window robustness analysis in Section 4.7 further evaluates this assumption under limited and contaminated calibration data.

At deployment initialization, the collected N_0 normal samples form the startup window W_0, which is used only for AdaBN-based anchor calibration, target-domain normal-reference estimation, and sample-level threshold initialization.

Based on W_0, the BN statistics of the source-domain reference model θ_{src} are recalibrated to obtain a frozen anchor model:

$$\theta_{\mathrm{anc}} = \mathrm{CalibBN}(\theta_{\mathrm{src}}, W_0), \tag{17}$$

where $\mathrm{CalibBN}(\cdot)$ denotes a forward-only calibration procedure that updates only the running mean and running variance of BN layers using W_0, without modifying the weights of fully connected layers.

The online model is initialized as

$$\theta_0 = \theta_{\mathrm{src}}. \tag{18}$$

Meanwhile, in the low-dimensional feature space, the mean vector and covariance matrix of the target-domain normal reference are estimated from W_0 as

$$\mu_T = \frac{1}{N_0}\sum_{i=1}^{N_0}\Phi\left(x_i^{T,0}\right), \tag{19}$$

and

$$\Sigma_T = \mathrm{Cov}\left(\left\{\Phi\left(x_i^{T,0}\right)\right\}_{i=1}^{N_0}\right). \tag{20}$$

These statistics are used later for sample-level protection in high-risk stages.

Next, W_0 is partitioned into calibration mini-batches $\left\{\mathcal{B}_k^{(0)}\right\}_{k=1}^{K_0}$. Here, K_0 is determined by the startup-window size and the online batch size. The hyperparameter adabn_calib_batches is implemented as a maximum calibration cap $K_{\max}$, rather than a requirement that exactly $K_{\max}$ mini-batches must exist. The effective number of AdaBN calibration mini-batches is therefore

$$K_{\mathrm{cal}} = \min(K_{\max}, K_0). \tag{21}$$

In the main setting, $N_0 = 1000$ and the online batch size is 256, so $K_0 = \lceil 1000/256 \rceil = 4$. Thus, when $K_{\max} = 20$, AdaBN calibration uses all available four startup mini-batches.

For the k-th calibration batch, its log-domain batch volatility is defined as

$$v_k = \frac{1}{m}\sum_{\ell=1}^{m} \mathrm{Var}_{x\in\mathcal{B}_k^{(0)}}\left(f_\ell^{\log}(x)\right), \tag{22}$$

where $f_\ell^{\log}(x)$ denotes the log-smoothed value of the ℓ-th retained feature for sample x.

Accordingly, the volatility baseline of the target-domain normal window is estimated by

$$\mu_v = \frac{1}{K_0}\sum_{k=1}^{K_0} v_k, \qquad \sigma_v = \sqrt{\frac{1}{K_0}\sum_{k=1}^{K_0}(v_k-\mu_v)^2}. \tag{23}$$

In addition, based on the Mahalanobis-distance distribution of samples in W_0 to the target-domain normal-reference center, a quantile threshold τ_{out} for strong-outlier protection and a score threshold τ_s for subsequent selective fallback are further estimated.

3.3.2 Entropy-Distribution Shift Detection

In the online stage, target-domain traffic arrives batch by batch. For the current batch defined in Eq. (7), the predictive entropy of each sample is first computed using the current online model θ_t, and its empirical cumulative distribution function $\widehat{F}_t(h)$ is constructed as in Eq. (9). This empirical distribution is then compared with the source-domain normal-entropy baseline $F_H(h)$, and the Kolmogorov–Smirnov (KS) statistic is used to quantify their discrepancy, namely, the shift measure Δ_t defined in Eq. (10).

When Δ_t is small, the current batch remains close to the source-domain normal reference in terms of model-uncertainty structure. Conversely, a large Δ_t indicates that the current input has significantly deviated from the normal reference distribution. Therefore, Δ_t serves as evidence for identifying potential domain change during the online target-domain stage.

3.3.3 SafeBrake Risk Gatekeeping

Relying solely on entropy-distribution shift detection may still be affected by local noisy samples and short-term abnormal fluctuations. To improve decision robustness, we further introduce a batch-volatility statistic. Let the log-domain value of the ℓ-th retained feature for the j-th sample in the current batch $\mathcal{B}_t$ be denoted by $f_{t,j}^{(\ell),\log}$. The average log-domain volatility of the current batch is defined as

$$v_t = \frac{1}{m}\sum_{\ell=1}^{m} \mathrm{Var}_{j=1,\ldots,b_t}\left(f_{t,j}^{(\ell),\log}\right), \tag{24}$$

where m is the dimension of the retained low-dimensional feature space, and $\mathrm{Var}_{j=1,\ldots,b_t}(\cdot)$ denotes the sample variance over the current batch.

Based on the normal volatility baseline (μ_v, σ_v) in Eq. (23), the SafeBrake risk-gating rule is defined as

$$\mathrm{State}_t = \begin{cases} \text{NoShift}, & \Delta_t \le \tau_{\text{KS}}, \\ \text{LowRiskShift}, & \Delta_t > \tau_{\text{KS}} \text{ and } v_t \le \mu_v + \alpha\sigma_v, \\ \text{HighRiskShift}, & \Delta_t > \tau_{\text{KS}} \text{ and } v_t > \mu_v + \alpha\sigma_v, \end{cases} \tag{25}$$

where τ_{KS} is the entropy-distribution shift threshold, and α is the volatility adjustment coefficient. When significant entropy shift and abnormal batch volatility occur simultaneously, the current batch is judged as

a high-risk shift and SafeBrake is triggered. If a significant shift is detected but the batch volatility remains within the normal range, the batch is judged as a low-risk shift. If Δ_t does not exceed the threshold, the batch is regarded as remaining in a no-shift state.

3.3.4 Constrained Online Update and Budgeted Selective Protection

When the current batch is determined to have a low-risk deviation, we adopt a restricted online update strategy to improve adaptation to target-domain data. Specifically, only the learnable affine parameters of the BN layers are updated, while all other weights are frozen. Let the set of learnable BN parameters at time t be denoted by $\Theta_{\mathrm{BN},t}$. The optimization objective is defined as

$$L_t = L_{\mathrm{align}} + \lambda_p L_{\mathrm{prior}} + \lambda_b L_{\mathrm{reg}}, \tag{26}$$

where L_{align} is the entropy-quantile alignment term, L_{prior} is the attack-ratio prior constraint, L_{reg} is the BN-parameter regularization term, and λ_p and λ_b are weighting coefficients.

The entropy-quantile alignment term is defined as

$$L_{\mathrm{align}} = \frac{1}{Q}\sum_{k=1}^{Q}\left(\hat{e}_{t,k} - \hat{e}_k\right)^2, \qquad \hat{e}_{t,k} = \widehat{F}_t^{-1}(q_k),\ \hat{e}_k = F_H^{-1}(q_k),\ q_k \in \{0.1, 0.2, \ldots, 0.9\}, \tag{27}$$

where Q denotes the number of selected quantile points, and $\widehat{F}_t^{-1}(\cdot)$ and $F_H^{-1}(\cdot)$ denote the inverse distribution functions of the current-batch entropy distribution and the source-domain normal-entropy baseline, respectively.

The attack-ratio prior constraint is defined as

$$L_{\mathrm{prior}} = \left(\hat{\pi}_t - \pi_a\right)^2, \qquad \hat{\pi}_t = \frac{1}{b_t}\sum_{j=1}^{b_t} P_{\theta_t}\left(y = 1 \mid \Phi\left(x_{t,j}^T\right)\right), \tag{28}$$

where $\hat{\pi}_t$ is the average attack risk of the current batch, and π_a is the attack-ratio prior. The prior π_a is not used as a direct estimate of the true attack proportion in the current batch. Instead, it serves as a conservative regularizer in the low-risk update objective, preventing systematic underestimation of attack probabilities under unlabeled conditions.

The BN-parameter regularization term is defined as

$$L_{\mathrm{reg}} = \sum_{\ell \in \mathcal{L}_{\mathrm{BN}}}\left(\|\gamma_{\ell,t}\|_2^2 + \|\beta_{\ell,t}\|_2^2\right), \tag{29}$$

where $\mathcal{L}_{\mathrm{BN}}$ denotes the set of all BN layers, and $\gamma_{\ell,t}$ and $\beta_{\ell,t}$ are the scale and shift parameters of the ℓ-th BN layer, respectively.

Accordingly, the BN-parameter update under low-risk conditions is written as

$$\Theta_{\mathrm{BN},t+1} = \Theta_{\mathrm{BN},t} - \eta \nabla_{\Theta_{\mathrm{BN},t}} L_t, \tag{30}$$

where η is the learning rate.

When the current batch is judged as a high-risk shift, online updating is suspended, i.e.,

$$\Theta_{\mathrm{BN},t+1} = \Theta_{\mathrm{BN},t}. \tag{31}$$

Under high-risk conditions, the system enters a protection mode. Based on the calibrated anchor model θ_{anc} and the target-domain normal-reference statistics, budgeted selective protection is performed for the current batch. First, the Mahalanobis distance from a sample to the target-domain normal-reference center is computed as

$$d_M(x) = \sqrt{(\Phi(x) - \mu_T)^\top \Sigma_T^{-1} (\Phi(x) - \mu_T)}, \tag{32}$$

where μ_T and Σ_T are given by Eqs. (19) and (20), respectively.

For strongly outlying samples, the anchor-model output is directly used instead. Let the strong-outlier set be

$$\mathcal{O}_t = \{x \in \mathcal{B}_t \mid d_M(x) \geq \tau_{\text{out}}\}, \tag{33}$$

where τ_{out} is the strong-outlier threshold. The attack risk given by the anchor model is defined as

$$r_{\text{anc}}(x) = P_{\theta_{\text{anc}}}(y = 1 \mid \Phi(x)). \tag{34}$$

For the remaining samples, a budgeted selective-fallback mechanism is further constructed. In the final configuration, only samples predicted as normal by the online model and not identified as strong outliers are considered for arbitration. The candidate set is therefore defined as

$$\mathcal{C}_t = \{x \in \mathcal{B}_t \mid r_t(x) < 0.5,\ d_M(x) < \tau_{\text{out}}\}. \tag{35}$$

To measure the necessity of correcting the current sample by the anchor model, the following score is defined:

$$s_t(x) = a\ \text{rank}_{01}(\text{logit}(r_{\text{anc}}(x))) + b\ \text{rank}_{01}(\text{logit}(r_{\text{anc}}(x)) - \text{logit}(r_t(x))) + c\ \text{rank}_{01}(d_M(x)), \tag{36}$$

where $\text{rank}_{01}(\cdot)$ denotes rank-based normalization to the interval $[0,1]$ within the candidate set, $\text{logit}(p) = \log\big(p/(1-p)\big)$, and a, b, and c are weighting coefficients.

The fallback-eligible subset and the selective-fallback budget are then defined as

$$\begin{aligned} \widetilde{\mathcal{C}}_t &= \{x \in \mathcal{C}_t \mid r_{\text{anc}}(x) \geq 0.5,\ s_t(x) \geq \tau_s\}, \\ k_t &= \lceil \rho\, |\mathcal{C}_t| \rceil, \end{aligned} \tag{37}$$

where τ_s is the score threshold estimated from the calibration window W_0, and $\rho \in (0,1)$ is the budget ratio for selective fallback.

Accordingly, the selective-fallback set is defined as

$$\mathcal{R}_t = \text{TopK}\big(\widetilde{\mathcal{C}}_t, s_t, k_t\big), \tag{38}$$

where $\text{TopK}(\cdot)$ selects the top k_t samples in descending order of $s_t(x)$.

Under the protection mechanism, the final attack risk is defined as

$$\widetilde{r}_t(x) = \begin{cases} r_{\text{anc}}(x), & x \in \mathcal{O}_t \cup \mathcal{R}_t, \\ r_t(x), & \text{otherwise}, \end{cases} \tag{39}$$

and the protected task-level trust score and class output are written as

$$\widetilde{T}_t(x) = 1 - \widetilde{r}_t(x), \qquad \widetilde{y}_t(x) = \mathbb{I}[\widetilde{r}_t(x) \geq 0.5]\,. \tag{40}$$

Here, $\mathbb{I}(\cdot)$ is the indicator function. In the binary setting considered in this paper, $\widetilde{y}_t(x) = 1$ denotes attack, while $\widetilde{y}_t(x) = 0$ denotes normal.

This mechanism ensures that strongly anomalous samples are preferentially protected by the anchor model, while only a limited number of high-risk samples with model disagreement are selectively corrected under a fixed budget. In this way, the spread of erroneous self-adaptation can be suppressed while the online model still retains its capability to recognize anomalous behavior. For ease of implementation, the startup-window calibration, entropy-shift detection, risk judgment, and constrained maintenance procedures are summarized in Algorithm 1.

Algorithm 1: RaL-TTA: risk-aware online trust maintenance algorithm

Input: Source reference model θ_{src}, source-domain normal-entropy baseline $F_H(h)$, target-domain normal calibration window W_0, target-domain online batch sequence $\{B_t\}_{t=1}^{T}$, entropy shift threshold τ_{KS}, volatility adjustment coefficient α, attack ratio prior π_a, weight coefficients λ_p, λ_b, learning rate η, selective fallback budget ρ, arbiter weights a, b, c

Output: Final trust scores $\{\tilde{T}_t(x)\}_{t=1}^{T}$ for each online batch sample

Initialize online model $\theta_0 \leftarrow \theta_{\text{src}}$

Initialize frozen anchor model $\theta_{\text{anc}} \leftarrow \text{CalibBN}(\theta_{\text{src}}, W_0)$ (see Eq. (17))

Estimate target-domain normal-reference statistics μ_T, Σ_T based on W_0 (see Eqs. (19) and (20))

Estimate batch volatility baseline (μ_v, σ_v) and sample-level threshold τ_{out} along with selective-fallback score threshold τ_s based on W_0

for $t = 1, 2, \ldots, T$ **do**

  Use online model θ_t to perform forward inference on current batch B_t, obtaining attack probability $r_t(x)$, original trust score $T_t(x)$, and predicted entropy $h_t(x)$ according to Eqs. (4)–(6)

  Construct empirical entropy distribution $\hat{F}_t(h)$ for the current batch via Eq. (9), and compute entropy-distribution shift Δ_t via Eq. (10)

  Calculate current batch volatility statistic v_t via Eq. (24), and determine current batch state State_t via Eq. (25)

  if $State_t = NoShift$ **then**

    Determine no significant shift occurs in current batch, set $\tilde{T}_t(x) \leftarrow T_t(x)$ and $\theta_{t+1} \leftarrow \theta_t$

  else

    if $State_t = HighRiskShift$ **then**

      Trigger SafeBrake, pause online update, and set $\theta_{t+1} \leftarrow \theta_t$ via Eq. (31)

      Perform inference on B_t using anchor model θ_{anc}, obtaining anchor-model attack probability $r_{\text{anc}}(x)$ via Eq. (34)

      Compute sample-level Mahalanobis distance via Eq. (32), and identify outlier sample set O_t via Eq. (33)

      Construct arbiter candidate set C_t via Eq. (35), and compute divergence score $s_t(x)$ via Eq. (36)

      Filter fallback-eligible set $\tilde{C}_t$ via Eq. (37), and construct selective fallback set R_t via Eq. (38)

      Obtain protected risk probability $\tilde{r}_t(x)$ and final trust score $\tilde{T}_t(x)$ via Eqs. (39) and (40)

    else

      Determine current batch is low-risk shift, set $\tilde{T}_t(x) \leftarrow T_t(x)$

(Continued)

Algorithm 1 (continued)

Construct online adaptive target and perform low-risk BN constrained update via Eqs. (26)–(30)
Only update learnable parameters $\Theta_{BN,t}$ of BN layers in online model, keep other parameters unchanged, and obtain θ_{t+1}
end
end
end

4 Experiments and Results Analysis

4.1 Experimental Environment and Datasets

4.1.1 Experimental Scenario and Cross-Domain Task

The experiments are conducted on the `DNN-EdgeIIoT-dataset.csv` file from the Edge-IIoTset benchmark proposed by Ferrag et al. [21]. This dataset is collected from network traffic in IIoT environments and contains both normal-behavior samples and multiple categories of attack samples, thereby providing a rich set of traffic features for edge-side security evaluation. In this study, the binary label `Attack_label` provided by the dataset is adopted as the supervision signal, where `Normal` is encoded as 0 and `Attack` is encoded as 1. The attack-type label `Attack_type` is used only for cross-domain task construction, statistical analysis, and result presentation, and is not involved in model input or online parameter updates. The overall statistics of the dataset are summarized in Table 2.

Table 2: Statistical summary of the dataset.

Category	Value
Total number of samples	2,219,201
Feature dimension	63
Number of normal samples	1,615,643
Number of attack samples	603,558
Number of attack categories	14

This paper focuses on the cross-domain behavior distribution shift problem faced by IIoT edge nodes under continuous online streaming conditions. Unlike protocol-only domain definitions, this study defines the cross-domain task as a compound-shift scenario involving both device communication relationships and attack types. In this setting, the normal behavior patterns and attack compositions differ between the source and target domains. The specific data division, online stream construction, and feature configuration are respectively presented in Sections 4.1.2 and 4.1.3.

4.1.2 Data Construction and Streaming Evaluation Settings

To ensure fair comparisons among different methods, ablation experiments, and parameter sensitivity experiments, this paper uniformly completes data partitioning, independent startup-window construction, and three-phase online stream pre-generation under the condition of fixed random seeds. Except for the variables under investigation, all experiments reuse the same data partitioning and streaming sequences. The source-domain training set is composed of source-domain normal traffic and source-domain attack samples extracted according to a fixed attack ratio. The normal and attack traffic of the target domain are split at the

group level and divided into validation and test sets in a 1:1 ratio to avoid instance-level leakage. The resulting dataset compositions are summarized in Tables 3 and 4.

Table 3: Composition of the source-domain training dataset.

Item	Description	Count
Source-domain normal samples	Non-MQTT normal traffic from the communication pair `192.168.0.101` ↔ `192.168.0.128`	1,000,310
Source-domain attack pool	Total number of source-domain attack samples from 11 attack categories	315,492
Offline attack ratio r_s	Fixed setting	0.2
Sampled attack instances	Drawn from the source-domain attack pool with $r_s = 0.2$	250,077
Total size of source-domain training set	Sum of normal samples and sampled attack samples	1,250,387

Table 4: Composition of the validation and test sets in the target domain.

Subset	Source of Normal Samples	Source of Attack Samples	# Normal Samples	# Attack Samples
Validation set	Group-wise split from the target-domain normal pool	Group-wise split from the target-domain attack pool	141,625	144,033
Test set	Group-wise split from the target-domain normal pool	Group-wise split from the target-domain attack pool	141,625	144,033
Total	Remaining non-MQTT normal traffic	`DDoS_TCP`, `DDoS_UDP`, `DDoS_ICMP`	283,250	288,066

Note: The source-domain attack samples, validation-set attack samples, and test-set attack samples are mutually exclusive. The normal samples in the validation set and the test set are also mutually exclusive.

Before online evaluation, an independent subset W_0 is drawn from the pool of target-domain normal samples to construct the deployment startup window, where $W_0 = 1000$ in the main experiments. This startup window is used only for AdaBN anchor calibration, target-domain normal-reference statistics estimation, and threshold initialization, and does not participate in the subsequent three-phase online evaluation. The online stream consists of an initial normal phase (Phase 1), a perturbation phase (Phase 2), and a recovery phase (Phase 3). The main experiments are conducted on the target-domain test split with settings $k = 8$, $r = 0.3$, $\pi_a = 0.1$, and sample-level arbitration budget $\rho = 0.01$. The main comparison adopts the random-mixing stream with $r = 0.3$, while the extended attack-ratio and burst-injection settings in Table 5 are retained as robustness-oriented implementation settings. In the five-seed main comparison, different random seeds are used to construct or evaluate pre-built target streams; for the burst-injection robustness

setting, the seed controls the attack-injection offset in Phase 2. The three-phase streaming settings are listed in Table 5.

Table 5: Experimental settings for three-phase streaming evaluation.

Phase/Item	Setting
Deployment startup window W_0	1000 target-domain normal samples, used only for deployment initialization and excluded from the three-phase online evaluation
Phase 1	5000 target-domain normal samples
Total length of Phase 2	3000 samples
Phase 3	5000 target-domain normal samples
Attack ratio r in the main experiments	0.3
Composition of the perturbation phase in the main experiments	900 attack samples + 2100 normal samples
Extended settings for attack ratio	$r \in \{0.1, 0.3, 0.5, 1.0\}$
Mode 1	Random mixing: attack samples are randomly inserted into the normal stream
Mode 2	Burst injection: attack samples are inserted into the normal stream in contiguous segments
Robustness setting for burst scenarios	Random seed of attack-injection offset: 0–4
Total length of the online stream	13,000 samples

4.1.3 Feature Preprocessing and Input Feature Configuration

After data construction, the raw traffic features are uniformly preprocessed and configured, and all methods share the same preprocessing pipeline and input feature space. For numerical features, the log-compression strategy described in Section 3.2.1 is applied. The standardization parameters are estimated only on the source-domain training set. They are then fixed and applied to the source-domain validation set, target-domain validation set, target-domain test set, the startup window W_0, and subsequent online evaluation streams to avoid leakage of target-domain statistical information.

To reduce the model dependence on protocol identifiers, explicit identity information, and protocol-specific fields, the experiments remove the label field, time field, communication-object identifier fields, and various protocol-specific fields, while retaining only general statistical features, behavioral features, and transport-layer-related features. On this basis, following the LoFT-IIoT feature-selection strategy, candidate features are screened on the source-domain training set and the final input dimensionality is determined. Unless otherwise specified, the main experiments adopt $k = 8$ input features. The final retained features are listed in Table 6.

Table 6: Final set of selected input features.

No.	Feature Name	Feature Category	Description
1	`tcp.checksum`	Transport-layer checksum feature	Characterizes TCP packet checksum behavior
2	`tcp.dstport`	Port and connection feature	Characterizes the distribution pattern of destination ports
3	`tcp.flags`	Packet-control feature	Characterizes the combination pattern of TCP flags
4	`tcp.flags.ack`	Packet-control feature	Characterizes ACK response behavior
5	`tcp.len`	Length-statistics feature	Characterizes variations in TCP packet length
6	`udp.time_delta`	Temporal-behavior feature	Characterizes changes in UDP packet inter-arrival time
7	`tcp.connection.fin`	Connection-state feature	Characterizes TCP connection termination behavior
8	`udp.stream`	Session-association feature	Characterizes UDP flow-level session association information

The use of eight retained features is motivated by the trade-off among cross-domain robustness, lightweight edge-side deployment, and avoidance of shortcut learning, rather than by an assumption that eight features are universally sufficient for all IIoT scenarios. During feature construction, label fields, timestamps, communication-object identifiers, and protocol-specific application-layer fields are removed to reduce identity or protocol shortcuts. The retained features cover complementary low-level behavioral evidence, including transport-layer integrity, port and connection patterns, packet-control behavior, packet-length statistics, temporal behavior, connection-state information, and flow-level session association. These categories jointly characterize packet control, timing, length, and connection behavior that remain meaningful under heterogeneous service or protocol shifts. Therefore, the eight-dimensional representation provides sufficient behavioral evidence for the evaluated cross-domain robust dynamic trust task while keeping the TrustMLP model and online BN maintenance lightweight. We do not claim that the same eight features are universally optimal for every deployment; rather, they are selected as a conservative low-dimensional configuration for the studied cross-domain evaluation setting.

4.1.4 Evaluation Metrics and Experimental Environment

To comprehensively evaluate the proposed method under cross-domain continuous online streams, the evaluation criteria are organized into four aspects, namely, overall classification performance, phase-wise streaming behavior, trust-score calibration, and resource overhead. Specifically, the overall classification metrics are used to measure the general detection capability of the model; the phase-wise metrics are introduced to characterize the dynamic behavior of the model during the initial deployment stage, the perturbation stage, and the recovery stage; the calibration metrics are used to evaluate the probabilistic interpretability of the task-level trust score; and the resource-overhead metrics are adopted to assess the feasibility of the proposed method under edge-side deployment conditions. The definitions of all evaluation metrics are summarized in Table 7, while the experimental environment and the main parameter settings are listed in Table 8.

Table 7: Main evaluation metrics and their definitions.

Metric Category	Metric Name	Definition
Overall classification performance	True Positive Rate (TPR)	Correctly identified attack samples
	False Positive Rate (FPR)	Normal samples incorrectly classified as attacks
	F1-score	Harmonic balance between precision and recall
	Accuracy (Acc)	Correctly classified samples among all samples
Phase-wise streaming behavior	Phase-1 accuracy (P1_Acc)	Stability at the beginning of target-domain deployment
	Phase-2 accuracy (P2_Acc)	Overall performance during the perturbation stage
	Phase-2 true positive rate (P2_TPR)	Attack-detection ability during the perturbation stage
	Phase-2 false positive rate (P2_FPR)	False-alarm level on normal samples during the perturbation stage
	Phase-2 F1-score (P2_F1)	Overall classification balance during the perturbation stage
	Phase-3 accuracy (P3_Acc)	Steady-state recovery performance after perturbation
Trust-score calibration	Expected Calibration Error (ECE)	Consistency between the trust score and empirical correctness
	Brier score	Mean squared error of predicted probabilities
	Negative Log-Likelihood (NLL)	Fit between predicted probabilities and true labels
Resource overhead	Total runtime	Total time required to process the complete online stream
	Average batch latency	Online processing delay per batch
	Throughput	Number of samples processed per unit time
	Peak memory usage	Maximum memory overhead during execution
	Number of online trainable parameters	Scale of model adjustment during online maintenance

Table 8: Experimental environment and main parameter settings.

Category	Item	Setting
Hardware environment	CPU	Intel i7-11800H
	GPU	NVIDIA GeForce RTX 3080 Laptop GPU (16 GB)
	Memory	64 GB
Software environment	Operating system	Windows 11
	Python	3.7.4
	PyTorch	1.13.1
Offline training	Optimizer	Adam
	Learning rate	1×10^{-3}
	Batch size	256
	Input feature dimension k	8
	Number of training epochs	20
	Label-smoothing factor	0.05
Online stage	Online batch size	256
	Startup-window size W_0	1000 Target-domain normal samples in the main setting
	Entropy-shift threshold τ_{KS}	0.25
	Strong-outlier quantile τ_{out}	0.99
	Fallback score threshold τ_s	0.80
	Volatility coefficient α	2.0
	Prior attack ratio π_a	0.1
	Online adaptation learning rate η	5×10^{-3}
	Sample-level arbitration budget ρ	0.01
	BN regularization weight	10^{-4}
	Number of update steps per batch	3
	Mahalanobis gating quantile threshold	0.90/0.99
	AdaBN anchor calibration	Max. 20 startup mini-batches; capped by available W_0 samples; effective 4 in the main setting
	AdaBN momentum	0.2

4.1.5 External X-IIoTID Validation Protocol

To evaluate generalization beyond Edge-IIoTset, we additionally conduct external validation on the X-IIoTID dataset [22]. Unlike the Edge-IIoTset main experiment, the X-IIoTID experiment adopts a service-holdout cross-domain protocol. The source domain contains multiple source services, while the target domain consists of disjoint Modbus, MQTT, and WebSocket services. The target perturbation stage further

includes multiple attack families, including false data injection, MQTT cloud broker subscription, Modbus register reading, scanning vulnerability, and fuzzing. This setting is used to evaluate whether RaL-TTA can maintain dynamic trust evaluation capability under external cross-service distribution shift.

4.2 Comparison of Cross-Domain Detection Performance

To compare the detection performance of different online adaptation strategies under cross-domain distribution shifts, we evaluate Source-Only, AdaBN-only [13], TENT [16], POEM [18], POEM+SafeBrake, and RaL-TTA under the unified experimental protocol described above. The results are reported as mean ± standard deviation over multiple pre-built target streams. The main Edge-IIoTset results are listed in Table 9. The RaL-TTA configuration uses the validation-selected hyperparameters described in Section 4.5.

Table 9: Comparison of different online adaptation strategies in the Edge-IIoTset perturbation and recovery stages.

Method	P2_TPR	P2_FPR	P2_F1	P2_Acc	P3_FPR	P3_Acc
Source-Only	1.0000 ± 0.0000	0.9000 ± 0.0054	0.4878 ± 0.0015	0.3700 ± 0.0037	0.8990 ± 0.0000	0.1010 ± 0.0000
AdaBN-only	1.0000 ± 0.0000	0.6016 ± 0.0135	0.5876 ± 0.0054	0.5789 ± 0.0094	0.6080 ± 0.0000	0.3920 ± 0.0000
TENT	0.2193 ± 0.0269	0.6774 ± 0.0124	0.1565 ± 0.0178	0.2916 ± 0.0120	0.6418 ± 0.0463	0.3582 ± 0.0463
POEM	0.9087 ± 0.0287	0.6884 ± 0.0391	0.5172 ± 0.0166	0.4907 ± 0.0272	0.7067 ± 0.0565	0.2933 ± 0.0565
POEM+SafeBrake	0.9993 ± 0.0006	0.0309 ± 0.0028	0.9649 ± 0.0032	0.9782 ± 0.0021	0.0332 ± 0.0000	0.9668 ± 0.0000
RaL-TTA	1.0000 ± 0.0000	0.0410 ± 0.0016	0.9544 ± 0.0017	0.9713 ± 0.0012	0.0352 ± 0.0000	0.9648 ± 0.0000

As shown in Table 9, Source-Only and AdaBN-only still suffer from high false-positive rates in the target domain, indicating that merely relying on the source-domain model or simple BN-statistics recalibration is insufficient to mitigate cross-domain mismatch. TENT and POEM behave unstably under continuous unlabeled streams, suggesting that online updates without explicit risk constraints are vulnerable to attack-contaminated drift. In contrast, RaL-TTA achieves a strong perturbation-stage trade-off against the external TTA baselines: it maintains a P2_TPR of 1.0000, reduces P2_FPR to 0.0410, and achieves a P2_F1 of 0.9544. The internal POEM+SafeBrake control obtains slightly lower P2_FPR and higher P2_F1 in this controlled stream, but it does not include the full task-level trust-score formulation and budgeted sample-level safeguard used by RaL-TTA. Therefore, the results should be interpreted as showing that risk-aware gating is essential for safe online maintenance, while the full RaL-TTA framework provides a conservative trust-evaluation design with only a small raw-performance cost relative to the strongest internal control.

Fig. 4 further visualizes the online behavior of different methods. Source-Only, TENT, and POEM show unstable or low rolling accuracy under the target-domain stream, whereas the risk-protected POEM-based variants maintain more stable trajectories. RaL-TTA preserves high rolling accuracy during the perturbation and recovery phases, which is consistent with the phase-wise metrics in Table 9. These trajectories further support the role of risk-aware gating and protection in stabilizing online trust evaluation under attack-contaminated target streams.

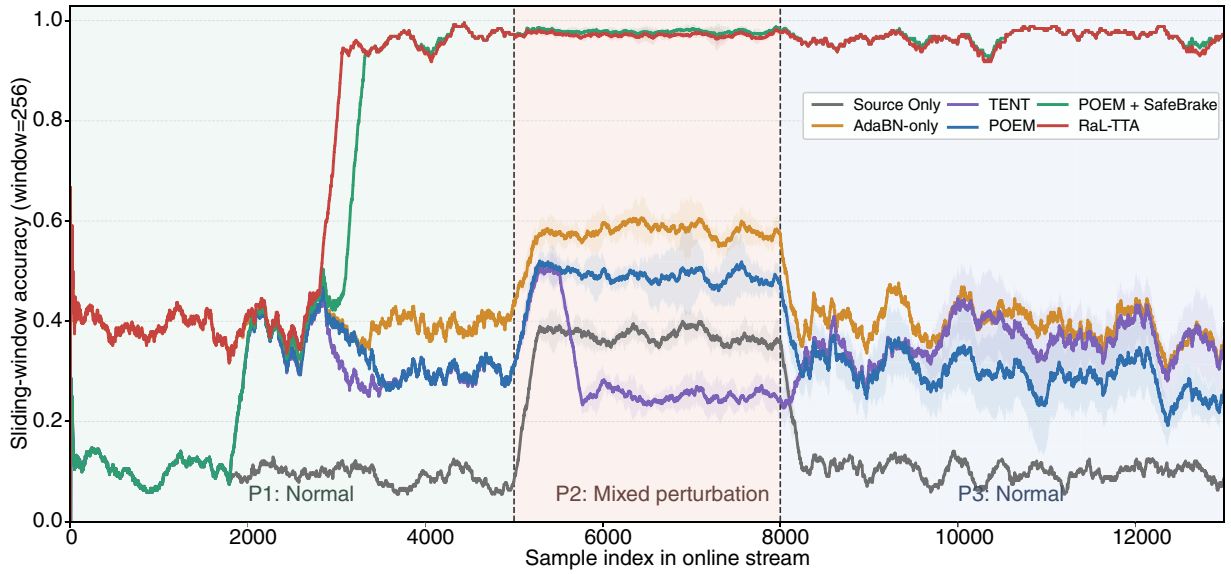

Figure 4: Sliding-window accuracy trajectories of different online adaptation methods in the three-phase continuous online stream. Each curve reports the mean accuracy over five pre-built target streams with a rolling window of 256 samples, and the shaded region indicates one standard deviation. The vertical dashed lines mark the transitions from the P1 normal phase to the P2 mixed-perturbation phase and from the P2 phase to the P3 normal recovery phase.

4.3 External Validation on X-IIoTID

To evaluate generalization beyond Edge-IIoTset, we further conduct external validation on X-IIoTID under the service-holdout protocol described in Section 4.1.5. Table 10 reports the overall representative results under the strong perturbation setting. Compared with Source-Only, RaL-TTA reduces P2_FPR and P3_FPR while maintaining meaningful attack recall. The external setting is more challenging than the Edge-IIoTset main setting because the target services and attack families are held out from the source domain.

Table 10: External X-IIoTID validation under the service-holdout strong perturbation setting.

Method	P1_Acc	P2_TPR	P2_FPR	P2_F1	P2_Acc	P3_FPR
Source-Only	0.7094	0.4144	0.3243	0.3818	0.5973	0.2938
AdaBN-only	0.8208	0.2944	0.0686	0.4049	0.7403	0.1798
TENT	0.8594	0.0222	0.0052	0.0430	0.7030	0.0156
POEM	0.8156	0.4011	0.1324	0.4691	0.7277	0.1516
POEM+SafeBrake	0.8184	0.3789	0.0986	0.4710	0.7447	0.1214
RaL-TTA	0.8388	0.3633	0.0952	0.4583	0.7423	0.1128

The X-IIoTID results show that general cross-domain TTA methods may produce very different trade-offs. For example, TENT obtains very low false-positive rates but almost loses attack recall in Phase 2. POEM-based variants achieve competitive P2_F1, whereas RaL-TTA provides lower post-perturbation false positives. Therefore, RaL-TTA should be understood as a risk-aware trust-maintenance framework that emphasizes the balance among attack detection, false-alarm suppression, and post-perturbation recovery rather than a method that maximizes every single metric.

Table 11 further reveals that the X-IIoTID service-holdout setting is challenging. RaL-TTA shows relatively stable behavior on MQTT-related target traffic, while Modbus exhibits larger seed-level variance and WebSocket remains difficult due to the mixture of normal and attack samples. These findings provide a fine-grained view of cross-service generalization and suggest that service-specific calibration remains an important direction for future work. Table 12 reports the per-attack-family recall of RaL-TTA under the X-IIoTID service-holdout setting.

Table 11: Fine-grained X-IIoTID per-service results of RaL-TTA in the target perturbation stage.

Target Service	Samples	Attack Samples	Normal Samples	P2_TPR	P2_F1
MQTT	360	360	0	0.4028 ± 0.0419	0.5734 ± 0.0429
Modbus	180	180	0	0.5426 ± 0.4651	0.6008 ± 0.5108
WebSocket	2460	360	2100	0.1806 ± 0.2152	0.1685 ± 0.1764

Table 12: -IIoTID per-attack-family recall of RaL-TTA.

Attack Family	Samples	Recall/TPR
False data injection	180	0.0926 ± 0.0463
MQTT cloud broker subscription	180	0.7130 ± 0.1297
Modbus register reading	180	0.5426 ± 0.4651
Scanning vulnerability	180	0.2111 ± 0.3368
Fuzzing	180	0.1500 ± 0.1164

The per-attack-family results indicate that different attack types have different degrees of cross-domain difficulty. MQTT cloud broker subscription attacks are detected more reliably, whereas false data injection, fuzzing, and scanning-related attacks remain more challenging. This observation is consistent with the difficulty of unlabeled external cross-service adaptation and is acknowledged as a limitation of the current framework.

4.4 Analysis of Key Mechanisms

4.4.1 Ablation Study of Key Modules

To analyze the main sources of performance improvement during the perturbation stage, this paper constructs three representative ablation settings: removing anchor protection (RaL-TTA w/o Anchor), removing budgeted sample-level safeguard/rollback (BSR; RaL-TTA w/o BSR), and always freezing updates (AlwaysFreeze). The results are summarized in Table 13.

Table 13 shows that simply freezing updates cannot effectively handle cross-domain abnormal perturbations because AlwaysFreeze preserves attack recall but causes a high false-positive rate and poor recovery. The variant without anchor protection is close to the POEM+SafeBrake control, indicating that SafeBrake-style risk gating is the dominant source of false-positive control in the Edge-IIoTset stream. The comparison between RaL-TTA and RaL-TTA w/o BSR further shows that budgeted sample-level rollback has only a limited numerical effect under the main setting. Thus, the ablation study supports a conservative interpretation: risk gating is the primary stabilization mechanism, whereas AdaBN anchor protection and

budgeted rollback provide additional safety boundaries for the full trust-evaluation framework rather than serving as the sole source of raw metric gains.

Table 13: Ablation results of key modules in the perturbation stage.

Method	P2_TPR	P2_FPR	P2_F1	P2_Acc	P3_Acc
AlwaysFreeze	1.0000 ± 0.0000	0.6016 ± 0.0121	0.5876 ± 0.0049	0.5789 ± 0.0084	0.3920 ± 0.0000
RaL-TTA w/o Anchor	0.9993 ± 0.0005	0.0309 ± 0.0025	0.9649 ± 0.0029	0.9782 ± 0.0019	0.9668 ± 0.0000
RaL-TTA w/o BSR	1.0000 ± 0.0000	0.0430 ± 0.0029	0.9522 ± 0.0031	0.9699 ± 0.0021	0.9648 ± 0.0000
RaL-TTA	1.0000 ± 0.0000	0.0410 ± 0.0015	0.9544 ± 0.0016	0.9713 ± 0.0010	0.9648 ± 0.0000

4.4.2 Analysis of Online Maintenance Behavior

To further illustrate how RaL-TTA operates in the main experiments, Table 14 summarizes the numbers of batches under different risk states and the corresponding maintenance actions across the three-phase online stream.

Table 14: Statistics of online maintenance states in the main experiment.

Phase	Total	No-Shift	Low-Risk	High-Risk	Update	Protect	Anchor	Rollback	Backup
Phase 1	20.0 ± 0.0	1.0 ± 0.0	15.0 ± 0.0	4.0 ± 0.0	12.0 ± 0.0	4.0 ± 0.0	4.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Phase 2	12.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	12.0 ± 0.0	0.0 ± 0.0	12.0 ± 0.0	12.0 ± 0.0	0.2 ± 0.4	0.0 ± 0.0
Phase 3	20.0 ± 0.0	11.0 ± 0.0	8.0 ± 0.0	1.0 ± 0.0	8.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Note: Update, Protect, Anchor, Rollback, and Backup denote update-executed, protection-mode, anchor-protected, rollback-covered, and backup-anchor actions, respectively.

Table 14 shows that all 12 batches in the perturbation stage are judged as high-risk and therefore enter the protection mode, with no online updates being executed. This indicates that the proposed method prioritizes freezing unreliable adaptation when attack traffic is mixed into the stream. In contrast, updates occur mainly in low-risk batches during the initial normal phase and the recovery phase, showing that RaL-TTA follows a selective online strategy of freezing at high risk and maintaining at low risk.

4.4.3 Analysis of Trust-Score Calibration Ability

To verify that the task-level trust scores output in this paper have probabilistic interpretability, we further assess calibration using expected calibration error (ECE), Brier score, and negative log-likelihood (NLL). Table 15 compares the trust-score reliability of RaL-TTA and representative baselines on the target-domain online stream.

Table 15 indicates that Source-Only suffers from large calibration errors after direct transfer to the target domain. During the perturbation stage, RaL-TTA and POEM+SafeBrake both provide substantially better calibration than Source-Only, with POEM+SafeBrake slightly lower on the three calibration metrics. Over the full stream, however, RaL-TTA obtains lower ECE, Brier score, and NLL, suggesting that the full protection-oriented trust-score output improves overall probabilistic reliability across deployment, perturbation, and recovery.

Table 15: Trust-score calibration results for different methods.

Evaluation Scope	Method	ECE	Brier Score	NLL
Perturbation stage	Source-Only	0.6000 ± 0.0024	0.5751 ± 0.0027	2.1230 ± 0.0145
Perturbation stage	POEM+SafeBrake	0.0483 ± 0.0024	0.0248 ± 0.0018	0.1217 ± 0.0061
Perturbation stage	RaL-TTA	0.0543 ± 0.0016	0.0298 ± 0.0011	0.1421 ± 0.0049
Full stream	Source-Only	0.8141 ± 0.0006	0.7649 ± 0.0006	2.8016 ± 0.0034
Full stream	POEM+SafeBrake	0.2354 ± 0.0004	0.1752 ± 0.0004	0.6213 ± 0.0014
Full stream	RaL-TTA	0.1896 ± 0.0003	0.1270 ± 0.0003	0.4161 ± 0.0011

4.5 Hyperparameter Selection and Sensitivity Analysis

To clarify how the key thresholds and online-adaptation parameters are selected, Table 16 summarizes the candidate values, selected values, and selection rules. The parameters are selected through a combination of validation-based tuning, startup-window quantile calibration, and conservative security-budget constraints.

Table 16: Hyperparameter and threshold selection procedure.

Parameter	Candidate Values/Range	Selected Value and Rule
τ_{KS}	0.15, 0.20, 0.25, 0.30, 0.35	0.25; Validation trade-off between drift sensitivity and false positives
τ_{out}	0.95, 0.97, 0.99, 0.995	0.99; Normal-window distance quantile for conservative outlier gating
τ_s/Selective-fallback score threshold	0.70, 0.75, 0.80, 0.85, 0.90	0.80; Validation low-FPR preference and startup-window score calibration
η/Online adaptation learning rate	10^{-3}, 2×10^{-3}, 5×10^{-3}, 10^{-2}	5×10^{-3}; Stable online adaptation
ρ/Arbitration budget	0, 0.005, 0.01, 0.02	0.01; Minimal nonzero conservative rollback budget
α/SafeBrake multiplier	1.5, 2.0, 2.5	2.0; Validation stability of SafeBrake activation
π_a	0.1, 0.2, 0.3, 0.5	0.1; Dev-set selection with P2_F1 and low-FPR preference
AdaBN calibration cap	5, 10, 20, all	Maximum 20 mini-batches; early stop when W_0 is exhausted; effective main setting: all 4 W_0 mini-batches

Table 17 shows that RaL-TTA remains stable under moderate variations of τ_{KS}, τ_{out}, τ_s, and the SafeBrake multiplier. The learning-rate analysis shows that an excessively small learning rate, such as 10^{-3}, fails to adapt sufficiently and leads to high false positives, whereas learning rates from 2×10^{-3} to 10^{-2} provide stable performance. The sensitivity analysis of ρ shows that rollback is not the dominant source of performance improvement under the main setting. A zero budget yields slightly lower FPR in this controlled stream, while a small nonzero budget such as 0.01 retains the conservative safeguard mechanism with only minor performance cost. The identical results for the AdaBN calibration caps of 5, 10, 20, and all are expected under the main startup-window setting. Since $W_0 = 1000$ and the online batch size is 256, only four startup

mini-batches are available. Therefore, any calibration cap no smaller than 5 is effectively equivalent to using all available startup samples.

Table 17: One-factor sensitivity analysis of major hyperparameters.

Parameter	Value	P2_TPR	P2_FPR	P2_F1	P3_FPR
τ_{KS}	0.15	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0352 ± 0.0000
τ_{KS}	0.20	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0352 ± 0.0000
τ_{KS}	0.25	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
τ_{KS}	0.30	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0338 ± 0.0000
τ_{KS}	0.35	1.0000 ± 0.0000	0.0440 ± 0.0028	0.9512 ± 0.0029	0.0350 ± 0.0000
τ_{out}	0.95	1.0000 ± 0.0000	0.0457 ± 0.0022	0.9494 ± 0.0023	0.0350 ± 0.0000
τ_{out}	0.97	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
τ_{out}	0.99	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
τ_{out}	0.995	1.0000 ± 0.0000	0.0406 ± 0.0028	0.9547 ± 0.0030	0.0344 ± 0.0000
τ_s	0.70	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
τ_s	0.75	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
τ_s	0.80	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
τ_s	0.85	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
τ_s	0.90	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
η	0.001	1.0000 ± 0.0000	0.5403 ± 0.0077	0.6134 ± 0.0034	0.3600 ± 0.0000
η	0.002	1.0000 ± 0.0000	0.0340 ± 0.0024	0.9619 ± 0.0026	0.0350 ± 0.0000
η	0.005	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
η	0.010	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0344 ± 0.0000
AdaBN cap	5	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
AdaBN cap	10	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
AdaBN cap	20	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
AdaBN cap	all	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
ρ	0	1.0000 ± 0.0000	0.0325 ± 0.0028	0.9634 ± 0.0030	0.0334 ± 0.0000
ρ	0.005	1.0000 ± 0.0000	0.0378 ± 0.0028	0.9578 ± 0.0030	0.0342 ± 0.0000
ρ	0.010	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
ρ	0.020	1.0000 ± 0.0000	0.0432 ± 0.0023	0.9520 ± 0.0024	0.0354 ± 0.0000
α	1.5	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
α	2.0	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0346 ± 0.0000
α	2.5	1.0000 ± 0.0000	0.0413 ± 0.0023	0.9541 ± 0.0024	0.0340 ± 0.0000

4.6 Computational Overhead Analysis

To compare the relative runtime cost of different methods, we conduct an overhead evaluation on a unified platform with a pre-built online stream. This subsection uses the test split under the pure-attack pressure setting $r = 1.0$ solely to compare the relative computational cost of different methods. For each online batch with batch size b and retained feature dimension m, the fixed MLP forward pass has linear cost in the batch size up to the constant hidden-layer widths. KS-based entropy comparison is performed on one-dimensional entropy values and costs $O(b \log b)$ if sorting is required. The Mahalanobis distance is computed in the selected low-dimensional feature space, where the inverse covariance matrix is precomputed from W_0, leading to $O(bm^2)$ per batch. Since only BN affine parameters are updated, the number of online trainable

parameters is 192 and the online update cost is independent of the full model size. The empirical overhead results are shown in Table 18 and Fig. 5.

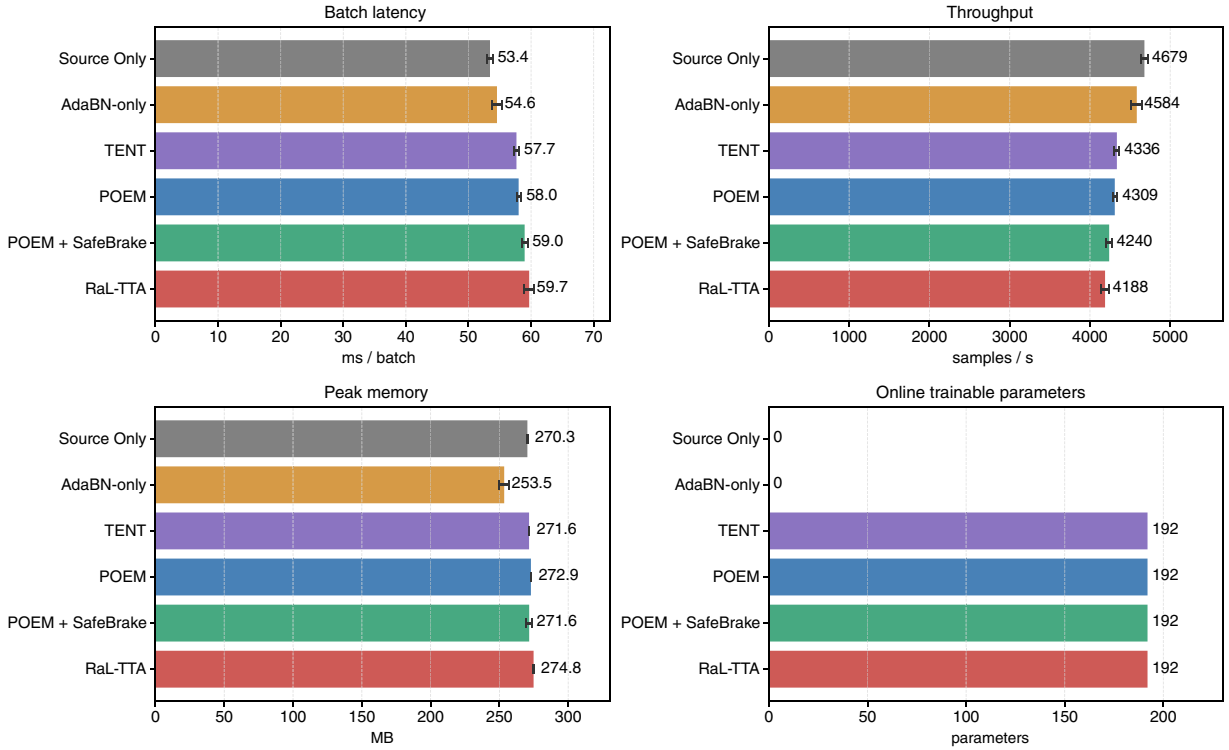

Figure 5: Overhead comparison of different online adaptation methods under a unified hardware and software platform with a pre-built online stream. The figure reports average batch latency, throughput, peak memory usage, and the number of online trainable parameters. Error bars indicate one standard deviation over three repeated measurements.

Table 18: Relative overhead comparison of different methods under a unified platform and a pre-built online stream.

Method	Runtime (s)	Latency (ms/Batch)	Throughput (Samples/s)	Memory (MB)	Trainable Params
Source-Only	2.7787 ± 0.0231	53.44 ± 0.44	4678.71 ± 38.70	270.34 ± 0.20	0
AdaBN-only	2.8368 ± 0.0424	54.55 ± 0.82	4583.61 ± 68.96	253.49 ± 3.57	0
TENT	2.9986 ± 0.0211	57.67 ± 0.41	4335.56 ± 30.43	271.57 ± 0.13	192
POEM	3.0167 ± 0.0165	58.01 ± 0.32	4309.47 ± 23.68	272.91 ± 0.22	192
POEM+SafeBrake	3.0666 ± 0.0254	58.97 ± 0.49	4239.50 ± 34.97	271.64 ± 2.11	192
RaL-TTA	3.1044 ± 0.0404	59.70 ± 0.78	4188.37 ± 54.22	274.84 ± 0.34	192

Note: Runtime, Latency, Memory, and Trainable params denote total runtime, average batch latency, peak memory usage, and the number of online trainable parameters, respectively.

From Table 18 and Fig. 5, RaL-TTA uses 192 online-trainable parameters, with a total runtime of 3.1044 s, an average batch latency of 59.70 ms/batch, a throughput of 4188.37 samples/s, and a peak memory usage of 274.84 MB. Its overhead is higher than Source-Only and basic TTA baselines because it performs risk gating and protected inference, but the cost remains bounded and the number of online trainable parameters

is unchanged at 192. These results support the feasibility of lightweight edge-side deployment while also clarifying that the additional safety mechanisms introduce a modest runtime and memory cost.

4.7 Startup-Window Robustness

The target-domain normal startup window is important for AdaBN calibration, target-domain normal-reference estimation, and sample-level threshold initialization. To evaluate the feasibility and limitation of this assumption, we test RaL-TTA under different startup-window sizes and contamination rates. This robustness experiment uses three seeds and independently resampled startup windows; therefore, the clean $W_0 = 1000$ setting is intended to evaluate robustness trends rather than exactly duplicate the five-seed main protocol in Table 9. The results are shown in Table 19 and Fig. 6.

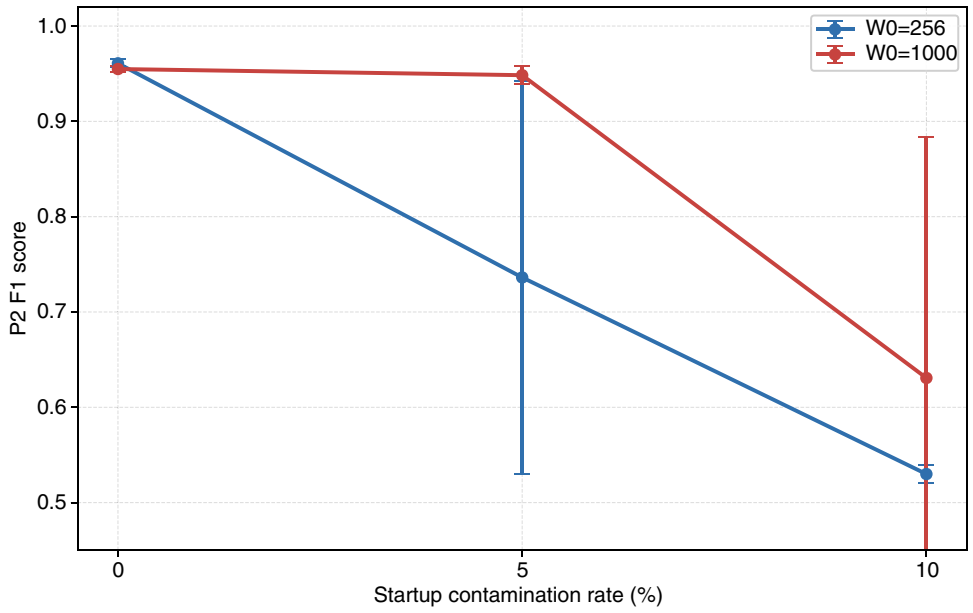

Figure 6: Startup-window robustness under contaminated startup windows. Curves report the perturbation-stage F1-score under different startup contamination rates for $W_0 = 256$ and $W_0 = 1000$. Error bars indicate one standard deviation over three random seeds.

Table 19: Robustness under limited and contaminated startup windows.

W_0 Size	Contamination	P2_TPR	P2_FPR	P2_F1	P3_FPR
256	0%	1.0000 ± 0.0000	0.0349 ± 0.0040	0.9609 ± 0.0043	0.0344 ± 0.0000
256	5%	0.6567 ± 0.3132	0.0265 ± 0.0167	0.7362 ± 0.2059	0.0225 ± 0.0125
256	10%	0.3700 ± 0.0068	0.0113 ± 0.0029	0.5300 ± 0.0093	0.0252 ± 0.0078
1000	0%	0.9881 ± 0.0023	0.0349 ± 0.0019	0.9549 ± 0.0032	0.0332 ± 0.0000
1000	5%	0.9870 ± 0.0034	0.0405 ± 0.0075	0.9484 ± 0.0093	0.0332 ± 0.0000
1000	10%	0.5526 ± 0.3148	0.0546 ± 0.0190	0.6308 ± 0.2523	0.0358 ± 0.0024

The results in Table 19 and Fig. 6 show that RaL-TTA performs reliably when the startup window is clean, even with 256 normal samples. With a sufficiently large startup window, the method also tolerates mild contamination. However, small contaminated windows or heavily contaminated startup data degrade

calibration reliability. This indicates that the normal startup window is a practical but nontrivial deployment assumption, and further robust initialization under contaminated startup conditions remains future work.

4.8 Discussion and Limitations

First, the trust score in this work is defined from a security-risk perspective. It is suitable for real-time edge-side security monitoring, but does not cover all dimensions of general trust management, such as long-term reputation, social interaction history, resource reliability, or quality-of-service evaluation. Second, RaL-TTA assumes that a short normal startup window is available for unsupervised calibration. Although this assumption is realistic during commissioning, maintenance restart, or trusted initialization, heavily contaminated startup data may weaken anchor construction and threshold estimation. Third, although this study includes external validation on X-IIoTID, both Edge-IIoTset and X-IIoTID are still public benchmark datasets. Real long-term industrial deployments may involve more complex temporal drift, device heterogeneity, and unseen attack behaviors. Fourth, false-positive control remains important for practical edge security systems. Although the main Edge-IIoTset setting reduces the perturbation-stage FPR to 0.0410, deployment-time alert fatigue still needs to be considered. In deployment, the trust score can be combined with multi-window smoothing, alert aggregation, or operator-confirmed escalation to reduce unnecessary alarms. Finally, the ablation results indicate that the conservative rollback branch has limited numerical effect under the main stream, and therefore more adaptive criteria for when to activate sample-level protection deserve further study.

5 Conclusion

This paper has proposed RaL-TTA, a risk-aware lightweight test-time adaptation (TTA) framework for security-oriented dynamic trust evaluation of IIoT edge nodes under cross-domain online streams. By combining a low-dimensional source-domain trust baseline, KS-based entropy-shift detection, SafeBrake risk gating, AdaBN anchor protection, and budgeted sample-level safeguards, RaL-TTA selectively maintains the online model under low-risk conditions and freezes unsafe adaptation under high-risk attack-contaminated streams. Experiments on Edge-IIoTset demonstrate that RaL-TTA improves perturbation-stage attack detection over general TTA baselines while substantially reducing false positives and maintaining post-perturbation stability. External validation on X-IIoTID further evaluates cross-service generalization across Modbus, MQTT, and WebSocket target services. Additional ablation, sensitivity, startup-window robustness, calibration, and overhead analyses show that the proposed method achieves a favorable balance among detection performance, trust-score reliability, adaptation safety, and edge-side efficiency. Future work will focus on more robust initialization under heavily contaminated startup windows, real edge-hardware deployment, and broader validation across long-term industrial traffic streams.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the National Natural Science Foundation of China [Grant No. 62102449] and the Science and Technology Research Project of Henan Province [Grant No. 252102211080].

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Qiuguo Guan and Zhiyu Ren; methodology, Qiuguo Guan; software and validation, Qiuguo Guan; formal analysis, Qiuguo Guan and Zhiyu Ren; writing—original draft preparation, Qiuguo Guan; writing—review and editing, Qiuguo Guan and Zhiyu Ren; supervision, Zhiyu Ren. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study are publicly available. Edge-IIoTset and X-IIoTID are available from their public dataset sources cited in the manuscript. The experimental code and processed scripts can be made available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable. This study does not involve human participants, human data, or animal experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Alotaibi B. A survey on industrial Internet of Things security: requirements, attacks, AI-based solutions, and edge computing opportunities. Sensors. 2023;23(17):7470. doi:10.3390/s23177470.
2. Liu DQ, Liang HL, Zeng XJ, Zhang Q, Zhang ZD, Li MH. Edge computing application, architecture, and challenges in ubiquitous power Internet of Things. Front Energy Res. 2022;10:850252. doi:10.3389/fenrg.2022.850252.
3. China Academy of Information and Communications Technology. White paper on Internet of Things (2020) [Internet]. Beijing, China: China Academy of Information and Communications Technology; 2020 [cited 2026 Mar 19]. Available from: http://www.caict.ac.cn/.
4. Ferraris D, Fernandez-Gago C, Roman R, Lopez J. A survey on IoT trust model frameworks. J Supercomput. 2024;80(6):8259–96. doi:10.1007/s11227-023-05765-4.
5. Garagad V, Iyer N. Dynamic trust-based device legitimacy assessment towards secure IoT interactions. J Commun Softw Syst. 2022;18(3):269–76. doi:10.24138/jcomss-2021-0189.
6. Motmi A, Alhazmi S, Abu-Khadrah A, Al-Akhras M, Alhosban F. Trust management in industrial Internet of Things using a trusted E-Lithe protocol. Int J Adv Comput Sci Appl. 2022;13(2):334–45. doi:10.14569/ijacsa.2022.0130239.
7. Jayasinghe U, Lee GM, Um TW, Shi Q. Machine learning based trust computational model for IoT services. IEEE Trans Sustain Comput. 2019;4(1):39–52. doi:10.1109/tsusc.2018.2839623.
8. Duque Anton SD, Sinha S, Schotten HD. Anomaly-based intrusion detection in industrial data with SVM and random forests. In: Proceedings of the 27th International Conference on Software, Telecommunications and Computer Networks (SoftCOM); 2019 Sep 19–21; Split, Croatia. Piscataway, NJ, USA: IEEE; 2019. p. 1–6.
9. Rabanser S, Günnemann S, Lipton ZC. Failing loudly: an empirical study of methods for detecting dataset shift. In: Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019); 2019 Dec 8–14; Vancouver, Canada. Red Hook, NY, USA: Curran Associates, Inc.; 2019. p. 1396–408.
10. Niu SC, Wu JX, Zhang YF, Wen ZQ, Chen YF, Zhao PL, et al. Towards stable test-time adaptation in dynamic wild world. In: Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023); 2023 May 1–5; Kigali, Rwanda.
11. Chen J, Mao FJ, Lv ZH, Tang JH. EdgeFD: an edge-friendly drift-aware fault diagnosis system for industrial IoT. In: Proceedings of the 2023 IEEE 23rd International Conference on Communication Technology (ICCT); 2023 Oct 13–16; Wuxi, China. Piscataway, NJ, USA: IEEE; 2023. p. 390–6.
12. Wang Q, Fink O, Van Gool L, Dai DX. Continual test-time domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. Piscataway, NJ, USA: IEEE; 2022. p. 7201–11.
13. Li YH, Wang NY, Shi JP, Hou XD, Liu JY. Adaptive batch normalization for practical domain adaptation. Pattern Recognit. 2018;80(3):109–17. doi:10.1016/j.patcog.2018.03.005.
14. Liang J, Hu DP, Feng JS. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of the 37th International Conference on Machine Learning (ICML); 2020 Jul 13–18; Virtual Event. New York, NY, USA: PMLR; 2020. p. 6028–39.
15. Liang J, He R, Tan T. A comprehensive survey on test-time adaptation under distribution shifts. Int J Comput Vis. 2025;133(1):31–64. doi:10.1007/s11263-024-02181-w.

16. Wang DQ, Shelhamer E, Liu ST, Olshausen BA, Darrell T. TENT: fully test-time adaptation by entropy minimization. In: Proceedings of the 9th International Conference on Learning Representations (ICLR 2021); 2021 May 3–7; Virtual Event.
17. Yuan YG, Xu BB, Hou L, Sun F, Shen HW, Cheng XQ. TEA: test-time energy adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 17–21; Seattle, WA, USA. Piscataway, NJ, USA: IEEE; 2024. p. 23901–11.
18. Bar Y, Shaer S, Romano Y. Protected test-time adaptation via online entropy matching: a betting approach. In: Proceedings of the Advances in Neural Information Processing Systems 37 (NeurIPS 2024); 2024 Dec 10–15; Vancouver, Canada. Red Hook, NY, USA: Curran Associates, Inc.; 2024. p. 85467–99.
19. Guan QG, Ren ZY, Wang QL. LoFT-IIoT: a lightweight trust feature extraction method for industrial Internet of Things. In: Proceedings of the 2025 IEEE 25th International Conference on Communication Technology (ICCT); 2025 Oct 16–18; Shenyang, China. Piscataway, NJ, USA: IEEE; 2025. p. 919–23.
20. Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML); 2017 Aug 6–11; Sydney, Australia. New York, NY, USA: PMLR; 2017. p. 1321–30.
21. Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access. 2022;10:40281–306. doi:10.1109/access.2022.3165809.
22. Al-Hawawreh M, Sitnikova E, Aboutorab N. X-IIoTID: a connectivity- and device-agnostic intrusion dataset for industrial Internet of Things. IEEE Internet Things J. 2022;9(5):3962–77. doi:10.1109/jiot.2021.3102056.