Black-box Backdoor Attack on Continual Semi-Supervised Learning for Time-series Internet-of-Things Systems

1st Thanh Cong Nguyen
Department of Computer Science
VNU - University of Engineering and Technology
Ha Noi, Vietnam
24025012@vnu.edu.vn

2nd Hanrui Wang§ and 3rd Isao Echizen
Information and Society Research Division
National Institute of Informatics
Tokyo, Japan
{hanrui_wang, iechizen}@nii.ac.jp

***Abstract*—Time-series Internet-of-Things (IoT) systems increasingly adopt continual semi-supervised learning (CSSL) to adapt to dynamic environments. However, the continual ingestion of unlabeled user data exposes such systems to backdoor injection threats. Existing backdoor attacks typically assume white-box access or manual control over labeled training data, assumptions that are often unrealistic in practical deployments. We present the first fully black-box backdoor attack against CSSL-based time-series classification that is gradient-free and annotation-free. Our approach constructs a surrogate model using only constrained query feedback from the target system. This surrogate is then used to optimize a trigger generator, enabling the injection of unlabeled poisoned samples that are automatically incorporated during CSSL updates. Under this challenging black-box setting, our attack achieves a high targeted attack success rate (average 78.4%) while inducing only a marginal drop in clean accuracy (average 2.5%). Moreover, the injected backdoor remains effective even when poisoning is performed only once every six system update rounds. These results demonstrate that CSSL pipelines are vulnerable under realistic black-box conditions, highlighting an urgent need for robust defenses.**

***Index Terms*—Backdoor attack, time-series classification, continual semi-supervised learning, internet of things.**

I. Introduction

Time-series data are ubiquitous in Internet-of-Things (IoT) applications, including health monitoring [1], human activity recognition [2], and energy management [3]. To sustain long-term performance under dynamic environments, many IoT systems adopt *continual semi-supervised learning* (CSSL), where models are periodically updated using pseudo-labels generated from streaming unlabeled user data [4], [5]. While CSSL improves adaptability, the continual ingestion of unlabeled inputs also introduces new security risks: a malicious adversary can inject carefully crafted backdoor samples into the data stream, causing the model to learn a hidden trigger that induces targeted misbehavior at inference time [6], as illustrated in Figure 1.

This work was partially supported by JSPS KAKENHI Grants JP21H04907 and JP24H00732, by JST CREST Grants JPMJCR20D3 and JPMJCR2562 including AIP challenge program, by JST AIP Acceleration Grant JPMJCR24U3, and by JST K Program Grant JPMJKP24C2 Japan.

§ Corresponding author.

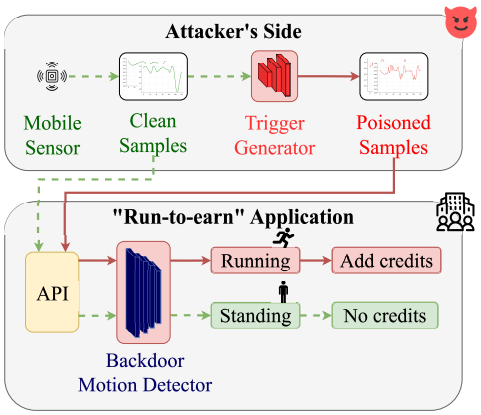


Fig. 1. An example of exploiting a backdoor in a "Run-to-Earn" application [2] to maliciously gain credits. An attacker with knowledge of the backdoor uses a trigger generator to manipulate input time-series data and mislead the system.

Despite growing interest in adversarial attacks on time-series IoT models, existing studies rely on assumptions that are misaligned with practical CSSL-based deployments, as summarized in Table I. First, a large body of work focuses on non-backdoor black-box adversarial attacks, which aim to induce misclassification at test time through carefully crafted perturbations, but do not implant persistent triggers that survive training or model updates [7]–[11]. Second, existing backdoor attacks typically assume white-box access, requiring full knowledge of the target model's architecture, parameters, or training procedure [12]–[16]. Third, most backdoor methods further rely on direct control over the training data, including manual annotation of poisoned samples with the target class, which is incompatible with unlabeled data streams in CSSL systems [17], [18]. Finally, both non-backdoor and backdoor attacks are generally designed for static training settings and do not account for continual semi-supervised updates, under which attack effectiveness often degrades rapidly if not repeatedly adapted [6], [19].

To address these challenges, we propose a gradient-free

TABLE I
COMPARISON OF ADVERSARIAL ATTACK METHODS FOR TIME-SERIES-BASED IOT MODELS, INCLUDING BACKDOOR AND BLACK-BOX ADVERSARIAL ATTACKS, AND THEIR APPLICABILITY TO CSSL SYSTEMS. [1] THIS THREAT MODEL HAS NOT BEEN SYSTEMATICALLY STUDIED IN PRIOR WORK; WE ADDRESS IT IN THIS PAPER.

Methodology	Backdoor Attack	Gradient Free	Annotation Free	Stable under Continual Updates[1]
TSFool [7] [2024]	✗	✗	✓	✗
Temporal TCA [8] [2025]	✗	✗	✓	✗
TSadv [9][2022]	✗	✓	✓	✗
BlackTreeS [10] [2023]	✗	✓	✓	✗
MCG [11] [2024]	✗	✓	✓	✗
TimeTrojan [12] [2022]	✓	✗	✗	✗
TSBA [13] [2023]	✓	✗	✗	✗
BACKTIME [14] [2024]	✓	✗	✗	✗
FreqBack [15] [2025]	✓	✗	✗	✗
TrojanTime [16] [2025]	✓	✗	✓	✗
Ultrasonic BD [17] [2022]	✓	✓	✗	✗
LIME-Guided [18] [2025]	✓	✓	✗	✗
Ours	✓	✓	✓	✓

and annotation-free fully black-box backdoor attack tailored to CSSL-based time-series IoT systems. Our approach first trains a surrogate model via knowledge distillation using only limited query feedback from the target system, without access to gradients or internal parameters. The surrogate is then used to optimize a trigger generator that produces poisoned samples. Instead of requiring manual annotation, these samples are injected through normal user interactions and are automatically pseudo-labeled and absorbed by the CSSL pipeline. Extensive experiments on two time-series datasets and three model architectures demonstrate that our attack achieves an average targeted attack success rate of 78.4% with only a 2.5% drop in clean accuracy. Moreover, once trained, the trigger generator remains effective even when updated only once every six system update rounds, substantially reducing the attacker's required effort.

Our main contributions are summarized as follows:

- We formalize the security risk of backdoor attacks against CSSL-based time-series IoT systems under a realistic fully black-box threat model.
- We propose a gradient-free and annotation-free attack pipeline that combines surrogate-based knowledge distillation with automated backdoor injection through pseudo-labeling.
- We conduct extensive evaluations across multiple datasets and models, demonstrating high attack success with minimal impact on clean accuracy and analyzing robustness under reduced update frequency.

II. RELATED WORKS

Non-backdoor Black-box Attacks on Time-series Systems. Adversarial attacks aim to induce misclassification by adding small perturbations to benign inputs [20]. Early black-box methods for time-series, such as TsAdv [9], relied on gradient-free optimization (e.g., differential evolution), followed by more query-efficient strategies like BlackTreeS [10]. MCG [11] further improved efficiency via meta-learning and transferability across surrogate models. Other works focus on stealthiness and task specificity, such as TsFool [7] for RNN-based classification and Temporal TCA [8] for time-series forecasting. These methods primarily target test-time evasion and do not implant persistent backdoors.

Backdoor Attacks on Time-series Systems. Backdoor attacks embed hidden malicious behaviors that are activated by specific triggers. TimeTrojan [12] first demonstrated backdoor attacks for time-series classification using gradient-based and evolutionary optimization, while TSBA [13] introduced sample-specific generative triggers to improve stealthiness. Subsequent work explored restricted-access or gradient-free settings, including ultrasonic triggers for speech recognition [17] and explanation-guided trigger placement [18]. Other directions include annotation-free backdoor training [16], frequency-domain attacks [15], and extensions to multivariate forecasting [14]. However, existing backdoor attacks typically assume white-box or gray-box access, static training, or manual data control, and do not address fully black-box backdoor injection under continual semi-supervised learning.

III. PRELIMINARIES

A. Time-Series Classification

In multivariate time-series classification, a dataset is defined as $\mathcal{D} = \{(X^{(i)}, y^{(i)})\}_{i=1}^{N}$, consisting of N labeled sequences. The objective is to learn a classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by θ, where the input space is $\mathcal{X} \subseteq \mathbb{R}^{T\times D}$ and the label space is $\mathcal{Y}$. Each input sample $X^{(i)} \in \mathbb{R}^{T\times D}$ represents a multivariate time series, where T denotes the sequence length (number of time steps) and D denotes the number of variables. Accordingly, $X^{(i)}$ can be decomposed as $X^{(i)} = \{x_1, x_2, \ldots, x_D\}$, where $x_d \in \mathbb{R}^T$ denotes the univariate time-series sequence of the d-th variable.

Deep neural networks (DNNs) are widely adopted for constructing the classifier f_θ due to their ability to model complex non-linear patterns in high-dimensional time-series data. Convolutional neural networks (CNNs) capture local

TABLE II
FOUR THREAT MODELS AND ATTACKER CAPABILITIES IN ACCESSING OR CONTROLLING THE TARGET SYSTEM.

Threat Models	Target Gradient	Training Dataset	Training Process
TM1: White BD.	✓	✓	✓
TM2: Gray BD.	✓	✗	✗
TM3: Black BD.	✗	✗	✗
TM4: Black Adv.	✗	✗	✗

temporal patterns through convolutional operations and progressively expand their receptive fields via stacked layers, enabling a global view of the sequence [21]. Alternatively, recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) [22] encode temporal dependencies sequentially according to $h_t = \text{RNN}(h_{t-1}, x_t)$. However, such recurrent architectures often suffer from gradient vanishing or explosion, which limits their ability to capture long-term dependencies. In contrast, Transformer-based models [23] employ attention mechanisms to directly model dependencies across the entire sequence, making them particularly effective for long-range temporal modeling. In this work, we focus on CNN-based and Transformer-based DNN architectures for time-series classification.

B. Continual Learning for IoT Systems

We consider a CSSL setting in which a service provider continuously collects new data and periodically updates its model by combining an existing trusted dataset with newly collected user data. To reduce annotation cost, pseudo-labels are generated automatically by the deployed model. Specifically, we assume that the service provider has trained an initial target model $\mathcal{M}_{\text{tgt}}$ to a satisfactory performance level using a trusted labeled dataset $\mathcal{D}_{\text{owner}}$. During deployment, unlabeled user data $\mathcal{D}_{\text{unlabeled}}$ are collected through normal query interactions. The target model assigns pseudo-labels to these samples to construct

$$\mathcal{D}_{\text{labeled}} = \{(x, \mathcal{M}_{\text{tgt}}(x)) \mid x \in \mathcal{D}_{\text{unlabeled}}\}, \tag{1}$$

which are then combined with the trusted dataset to form the training set $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{owner}} \cup \mathcal{D}_{\text{labeled}}$. The model is periodically retrained on $\mathcal{D}_{\text{train}}$ to produce updated versions that adapt to evolving data distributions.

C. Threat Model

We consider four threat models that differ in the attacker's ability to access the target model, training data, and training procedure, as summarized in Table II.

- **TM1: White-box Backdoor Attack.** In this strongest threat model, the attacker has full white-box access to the target system, including the model architecture, parameters, gradients, training dataset, and training process. The attacker can directly optimize backdoor triggers using gradient information and explicitly inject labeled poisoned samples into the training data. While widely adopted in prior backdoor studies [13]–[15], this setting assumes an unrealistic level of access for deployed IoT systems.
- **TM2: Gradient-access Backdoor Attack.** In this setting, the attacker has access to the target model's gradients but does not control the training dataset or training process. Compared to TM1, the attacker cannot directly modify the training data or labels, but can leverage gradient information to optimize backdoor triggers more efficiently. This threat model can be viewed as a relaxed variant of our black-box backdoor setting, where access to target gradients removes the need for surrogate model training.
- **TM3: Fully Black-box Backdoor Attack (Our Focus).** In this threat model, the attacker has no access to the target model's gradients, training dataset, or training process, and can only interact with the system through a standard query interface. We assume the query interface returns either (i) the top-k predicted classes with their confidence scores/probabilities, as commonly provided by commercial APIs, or (ii) only the top-1 label as a stricter variant.
  Poisoned samples must be introduced via normal user interactions and are incorporated into training solely through the CSSL pipeline. We assume the attacker does not receive privileged update notifications. In practice, model updates may be inferred from (i) public release notes/version announcements, or (ii) lightweight probing using a small fixed set of inputs and monitoring prediction drift. Importantly, exact update timing is not required for our attack: the attacker can refresh the trigger generator only occasionally (e.g., every few CSSL rounds) while retaining high effectiveness (see Section VII-A).
  To the best of our knowledge, this is the *first work* to systematically study backdoor attacks under this fully black-box and annotation-free threat model for CSSL-based time-series IoT systems.
- **TM4: Black-box Adversarial Attack.** This threat model corresponds to conventional black-box adversarial attacks that aim to induce misclassification at inference time using query-based perturbations. Unlike TM1–TM3, TM4 does not implant persistent triggers during training and does not address backdoor persistence under continual semi-supervised updates.

IV. FULLY BLACK-BOX BACKDOOR ATTACK

A. Problem Statement

Under the fully black-box threat model (TM3), the attacker aims to learn a trigger generator $\mathcal{G}(\cdot;\theta_g)$ that produces an input-dependent trigger pattern $\mathcal{G}(x;\theta_g)$ for any benign input x. A poisoned sample is constructed as

$$x' = x + \alpha \|x\|_\infty \mathcal{G}(x;\theta_g), \quad 0 \le \alpha \le 1, \tag{2}$$

where α controls the perturbation magnitude relative to the maximum amplitude of x, ensuring that the perturbation satisfies $\|x' - x\|_\infty \le \alpha \|x\|_\infty$. The attacker's objective is that, after each CSSL update, the updated target model $\mathcal{M}'_{\text{tgt}}$ incorporates the backdoor associated with $\mathcal{G}$, such that

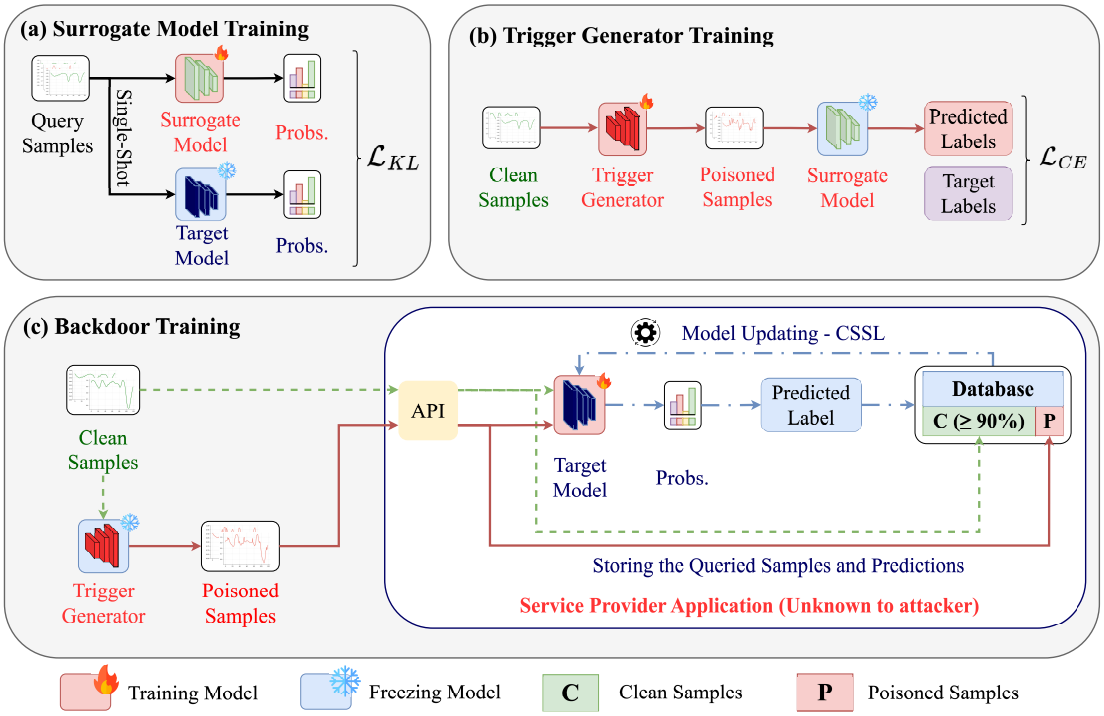


Fig. 2. Overview of the proposed fully black-box backdoor attack pipeline, including surrogate model training via query-based distillation, trigger generator optimization, poisoned sample injection, and progressive backdoor reinforcement through CSSL updates.

triggered inputs are classified into a predefined target class, i.e., $\mathcal{M}'_{\text{tgt}}(x') = y_{\text{tgt}}$, while benign inputs remain correctly classified as $\mathcal{M}'_{\text{tgt}}(x) = y_{\text{true}}$.

B. Surrogate Model Training

Due to the lack of gradient and parameter access, the attacker first trains a surrogate model $\mathcal{M}_{\text{sur}}$ to approximate the black-box target model $\mathcal{M}_{\text{tgt}}$, as illustrated in Figure 2(a). The attacker performs query-based knowledge distillation [11] by querying $\mathcal{M}_{\text{tgt}}$ on samples drawn from an auxiliary dataset $\mathcal{D}_{\text{atk}}$ and collecting the returned prediction probabilities to construct a distillation dataset $\mathcal{D}_{\text{distill}}$.

The distillation process is iterative. An initial batch of β_{init} queries is used to train a baseline surrogate model. In each subsequent round κ, which includes surrogate refinement and trigger generator training, an additional β_{inject} queries are issued and appended to $\mathcal{D}_{\text{distill}}$. To remain synchronized with the evolving target model, $\mathcal{D}_{\text{distill}}$ is cleared and rebuilt whenever $\mathcal{M}_{\text{tgt}}$ is updated by the service provider. The surrogate parameters θ_{sur} are optimized by minimizing the Kullback–Leibler (KL) divergence between the surrogate and target predictions:

$$\theta^*_{\text{sur}} = \arg\min_{\theta_{\text{sur}}} \mathbb{E}_{x\sim\mathcal{D}_{\text{distill}}}\left[\mathcal{L}_{\text{KL}}\left(\mathcal{M}_{\text{sur}}(x;\theta_{\text{sur}}), \mathcal{M}_{\text{tgt}}(x)\right)\right]. \quad (3)$$

Top-k outputs. In practice, black-box APIs may not expose a full probability vector over all classes, but instead return only the top-k predictions with confidence scores [24]. Our distillation remains applicable by constructing a sparse teacher distribution over the returned top-k classes and renormalizing it to sum to one; the surrogate is then trained to match this truncated distribution. Concretely, let $\mathcal{K}(x)$ denote the returned top-k class set for input x, and let $\tilde{p}_{\text{tgt}}(y \mid x)$ be the renormalized probabilities on $\mathcal{K}(x)$. We minimize $\mathcal{L}_{\text{KL}}(\mathcal{M}_{\text{sur}}(x), \tilde{p}_{\text{tgt}}(\cdot \mid x))$, which only requires the top-k scores. As a stricter variant, if only the top-1 label is available, the attacker can distill from hard labels by replacing the KL loss with standard cross-entropy to the predicted label [24].

C. Trigger Generator Training

Given the trained surrogate model $\mathcal{M}_{\text{sur}}$, the attacker optimizes a trigger generator $\mathcal{G}(\cdot;\theta_g)$, as illustrated in Figure 2(b). For each input x, the generator outputs a trigger pattern that is scaled by the relative perturbation magnitude $\alpha\|x\|_\infty$ and added to x to form the poisoned sample x' defined in Eq. (2). The generator parameters are optimized to induce the target misclassification on the surrogate model by minimizing the cross-entropy loss:

$$\begin{aligned} x' &= x + \alpha\|x\|_\infty \mathcal{G}(x;\theta_g), \\ \theta^*_g &= \arg\min_{\theta_g} \mathbb{E}_{x\sim\mathcal{D}_{\text{atk}}}\left[\mathcal{L}_{\text{CE}}\left(\mathcal{M}_{\text{sur}}(x'), y_{\text{tgt}}\right)\right], \end{aligned} \quad (4)$$

subject to the perturbation constraint $\|x' - x\|_\infty \le \alpha\|x\|_\infty$.

D. Backdoor Injection and Model Update

Once the trigger generator is trained, the attacker injects poisoned samples $\mathcal{D}_{\text{poison}}$ into the target IoT system by submitting them through normal query interfaces, as shown in Figure 2(c). These poisoned samples are indistinguishable from benign

user inputs. During the next CSSL cycle, the service provider constructs the unlabeled pool $\mathcal{D}_{\text{unlabeled}} = \mathcal{D}_{\text{benign}} \cup \mathcal{D}_{\text{poison}}$ and assigns pseudo-labels using the current target model to form $\mathcal{D}_{\text{labeled}} = \{(x, \mathcal{M}_{\text{tgt}}(x)) \mid x \in \mathcal{D}_{\text{unlabeled}}\}$. The target model is then updated by retraining on $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{owner}} \cup \mathcal{D}_{\text{labeled}}$:

$$\theta'_{\text{tgt}} = \arg\min_{\theta_{\text{tgt}}} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}} \left[\mathcal{L}_{\text{CE}}(\mathcal{M}_{\text{tgt}}(x;\theta_{\text{tgt}}), y)\right]. \quad (5)$$

Although some poisoned samples may initially receive correct pseudo-labels, successive CSSL updates progressively reinforce the backdoor effect, increasing the proportion of target-labeled poisoned samples and improving the attack success rate over time.

V. Experiment Settings

To support transparency and reproducibility, our code and datasets are publicly available at https://github.com/Lelvels/blackbox-bd-tsc.

A. Datasets

We evaluate the proposed attack on two public time-series datasets. **iAWE** [25] is a household power consumption dataset sampled at 1 Hz, containing approximately 4.7 million time-series instances. **MotionSense** [26] is a human activity recognition dataset collected from smartphone IMU sensors sampled at 50 Hz, with approximately 23 K instances. For both datasets, raw time-series data are segmented into fixed-length windows of 120 time steps. Each dataset is split into approximately 85% clean training data, which are assumed to be owned and trusted by the service provider, and 15% auxiliary data that are accessible to the attacker and used as the poison source.

B. Models and Training Settings

Target models. We consider three representative time-series classification architectures: Fully Convolutional Network (FCN) [21], ResNet [27], and Transformer [23]. The FCN and ResNet implementations follow the architectures proposed by [21]. The Transformer model consists of four encoder layers with four attention heads per layer and an attention head dimension of 128. ReLU is used as the activation function for all models. All target models are trained using the Adam optimizer with an initial learning rate of 10^{-3}. For the iAWE dataset, we use a batch size of 128 and train for at least 40 epochs. For the MotionSense dataset, we use a batch size of 64 and train for at least 200 epochs.

Surrogate model. A custom ResNet architecture is adopted as the surrogate model across all experimental settings to approximate the black-box target model.

Trigger generator. The trigger generator employs a dual-head 1D CNN architecture with a shared encoder composed of three convolutional blocks (kernel sizes 15, 21, and 8), each followed by batch normalization and ReLU activation. The shared encoder branches into a pattern head that produces the trigger waveform using a *tanh* activation and an amplitude head that regulates the trigger scaling. The generator is trained using the same optimizer and batch size settings as the corresponding target model. It is trained for 40 epochs on iAWE and 100 epochs on MotionSense, until the attack success rate on the target model converged.

TABLE III
Attack budget constraints. The reported poisoned rate denotes the proportion of poisoned samples in the data stream used for each CSSL update.

Dataset	Perturbation Magnitude (α)	Attack Epochs per Update	Max Number of Queries	Poisoned Rate
iAWE	0.3–0.5	50	20,000	0.68%
MotionSense	0.3–0.5	50	2,000	10%

C. Attack Budgets

To ensure attack stealthiness and practical feasibility, we impose explicit budget constraints on the attacker. Specifically, the trigger amplitude is bounded to limit perceptibility, the number of attack epochs per CSSL update is restricted, and the number of injected queries is constrained to remain a small fraction of the overall data stream used for model updates. These constraints define three key hyperparameters: the perturbation magnitude control α, the number of attack epochs per update, and the maximum number of injection queries. The detailed budget settings for each dataset are summarized in Table III.

All experiments are conducted on a single NVIDIA A100 GPU.

D. Evaluation Metrics

We evaluate **targeted** backdoor attacks using two metrics. **Attack Success Rate (ASR)** is defined as the percentage of poisoned samples that are classified into the target class, measuring attack effectiveness. **Clean Accuracy (CA)** denotes the classification accuracy on benign test samples after the attack, reflecting the model's ability to preserve normal functionality. A successful and stealthy backdoor attack achieves a high ASR while maintaining a high CA.

VI. Performance Evaluation

A. Overall Attack Effectiveness

Table IV reports the attack effectiveness under four threat models (TM1–TM4) across two datasets and three time-series classifiers, measured by ASR and CA. These threat models represent progressively restricted attacker capabilities, ranging from full white-box access to fully black-box interaction.

We first focus on **TM3 (fully black-box backdoor attack)**, which is the primary setting studied in this work. Under this most restrictive threat model, the attacker has no access to target gradients, training data, or training procedures, and can only interact with the system through standard query interfaces. Despite these limitations, TM3 consistently achieves high ASR across all datasets and models. On iAWE, TM3 attains a worst ASR of 83.3% with CA above 90% for all models. On MotionSense, TM3 achieves a worst ASR of 62.6% while preserving CA close to the clean baseline. These results demonstrate that persistent backdoors can be

TABLE IV

ATTACK EFFECTIVENESS UNDER FOUR THREAT MODELS (TMS) ACROSS DATASETS AND MODELS, REPORTED AS ATTACK SUCCESS RATE (ASR) AND CLEAN ACCURACY (CA). TM1 AND TM4 ADOPT REPRESENTATIVE EXISTING METHODS, WHILE TM2 AND TM3 CORRESPOND TO OUR PROPOSED BACKDOOR PIPELINE UNDER GRADIENT-ACCESS AND FULLY BLACK-BOX SETTINGS, RESPECTIVELY.

Datasets	**Models**	**ASR / CA (%)**			
		TM1: White BD. [13]	**TM2: Gray BD. (Ours)**	**TM3: Black BD. (Ours)**	**TM4: Black Adv.** [11]
iAWE [25]	FCN [21]	99.5 / 99.7	92.2 / 99.7	83.3 / 90.3	55.4 / 99.9
	Resnet [21]	99.1 / 98.2	95.8 / 98.2	92.7 / 94.9	50.8 / 98.3
	Transformer [23]	96.2 / 96.4	90.0 / 96.4	87.7 / 95.3	48.8 / 96.2
MotionSense [26]	FCN [21]	99.3 / 99.3	75.8 / 98.8	72.9 / 99.0	51.5 / 99.2
	Resnet [21]	99.1 / 99.4	77.3 / 99.0	71.1 / 98.6	52.5 / 98.6
	Transformer [23]	97.6 / 97.5	65.2 / 97.0	62.6 / 97.2	35.2 / 97.6

successfully implanted through unlabeled data streams alone, even under fully black-box CSSL settings.

TM2 (gradient-access backdoor attack) is a relaxed variant of TM3, where access to target gradients removes the need for surrogate model training. As shown in Table IV, TM2 consistently outperforms TM3 in terms of ASR while maintaining comparable CA. This behavior is expected, as gradient access provides more accurate optimization signals for trigger generation. The relatively small performance gap between TM2 and TM3 indicates that the proposed surrogate-based approach effectively approximates gradient information under fully black-box constraints.

For reference, **TM1 (white-box backdoor attack)** and **TM4 (black-box adversarial attack)** are evaluated using representative existing methods. TM1 achieves near-perfect ASR with negligible CA degradation due to its strong access assumptions. In contrast, TM4 exhibits substantially lower ASR across all settings, as black-box adversarial attacks are designed for test-time evasion and do not implant persistent backdoors that survive CSSL updates. Together, these results highlight that TM3 occupies a distinct and previously unexplored regime between overly strong white-box backdoor attacks and non-persistent black-box adversarial attacks.

B. Class-wise Attack Performance

To further analyze the robustness of the proposed attack, Table V presents class-wise ASR under TM2 and TM3. Across both datasets, the attack maintains consistently high ASR for most target classes, indicating that its effectiveness is not confined to a small subset of semantic categories. On iAWE, TM3 achieves strong ASR across all appliance classes. On MotionSense, although certain motion classes such as *Walk* exhibit lower ASR, the attack remains effective across diverse activity patterns. These results suggest that the proposed backdoor attack generalizes well across different classes and temporal dynamics, rather than overfitting to specific behaviors.

TABLE V

CLASS-WISE ATTACK SUCCESS RATE (ASR, %) OF THE PROPOSED BACKDOOR ATTACK UNDER TM2 (GRADIENT-ACCESS BACKDOOR) AND TM3 (FULLY BLACK-BOX BACKDOOR) SETTINGS.

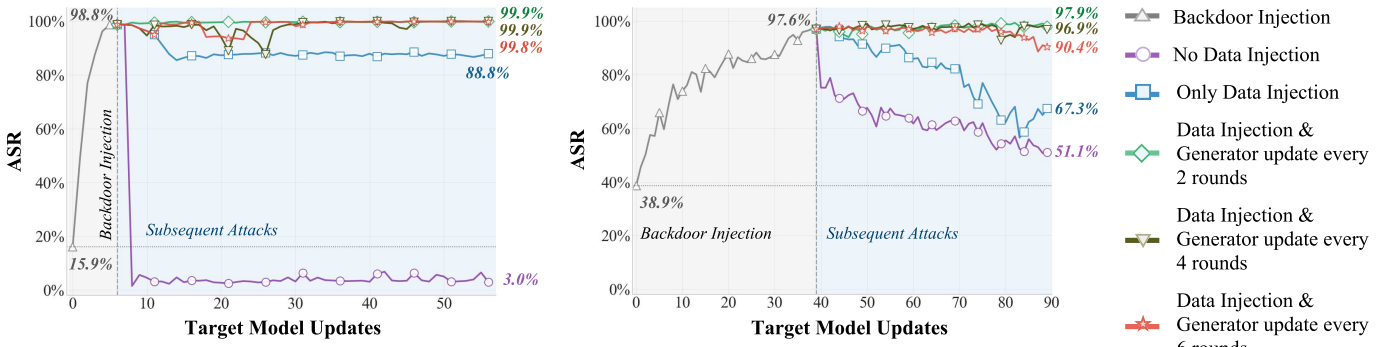
Dataset	**Class**	**TM2: Gray BD. / TM3: Black BD.**		
		FCN	**ResNet**	**Transformer**
iAWE	Air Con. 1	88.0 / 73.7	99.9 / 99.9	90.2 / 82.2
	Air Con. 2	80.0 / 73.8	96.7 / 73.8	99.4 / 99.3
	Computer	99.2 / 98.8	99.8 / 99.8	83.0 / 80.6
	Fridges	99.4 / 75.3	99.8 / 90.8	95.2 / 94.4
	Television	95.2 / 95.5	95.4 / 99.4	82.2 / 81.9
	Average	92.2 / 83.3	95.8 / 92.7	90.0 / 87.7
MotionSense	Downstairs	75.1 / 68.6	72.0 / 71.9	75.2 / 73.1
	Jog	73.5 / 73.4	72.3 / 71.3	50.3 / 50.8
	Sit	83.0 / 79.5	99.0 / 66.0	98.3 / 97.6
	Stand	75.0 / 72.4	77.0 / 73.8	75.4 / 66.9
	Upstairs	73.3 / 73.1	69.0 / 73.1	60.2 / 51.3
	Walk	75.0 / 70.5	75.0 / 70.5	31.6 / 35.7
	Average	75.8 / 72.9	77.3 / 71.1	65.2 / 62.6

VII. ABLATION STUDY

A. Backdoor Stability under CSSL Updates

We analyze the update efficiency and stability of backdoor attacks under CSSL by examining how frequently the attacker must refresh the trigger generator to maintain a high attack success rate. After the backdoor converges, we track ASR across successive CSSL updates under different generator update schedules. Specifically, we compare a *one-shot attack*, where no further poisoning is performed after initial backdoor implantation, with *delayed-update attacks*, in which the trigger generator is retrained and re-injected once every 2, 4, or 6 CSSL update rounds.

As shown in Figure 3, two key observations emerge. First, without continued poisoning (purple curves), the implanted backdoor is gradually overwritten by subsequent CSSL retraining, resulting in a rapid decline in ASR. Second, even with infrequent generator updates, the backdoor remains highly effective. In particular, updating the trigger generator only once every six rounds (red curves) preserves a high ASR, reaching 99.8% on iAWE and 90.4% on MotionSense. These results demonstrate that maintaining a persistent backdoor under CSSL does not require continuous attacker intervention, highlighting both the practicality and severity of the proposed fully black-box backdoor attack. This update efficiency also reduces the need for precise update detection, since the at-



(a) iAWE *(Target: FCN; All class → "computer")* (b) MotionSense *(Target: Transformer; All class → "sit")*

Fig. 3. Attack update efficiency under continual semi-supervised learning (CSSL). After the backdoor converges (gray curves), the trigger generator update frequency can be reduced from every update round to once every six rounds (red curves) while preserving high ASR. In contrast, without further poisoning, the implanted backdoor is progressively overwritten by subsequent CSSL updates, leading to a sharp ASR degradation (purple curves).

tacker can probe or refresh on a coarse schedule (e.g., every six rounds) while maintaining high ASR.

B. Effect of Perturbation Magnitude

We investigate the impact of perturbation magnitude control α on backdoor effectiveness under TM3. Figures 4 and 5 illustrate the relationship between α and the resulting ASR on the iAWE and MotionSense datasets, respectively.

As α increases, ASR improves rapidly at small perturbation budgets, indicating that even weak trigger signals can be amplified through CSSL updates. For both datasets, the attack performance stabilizes once α reaches approximately 0.4, beyond which further increases in perturbation magnitude yield only marginal gains in ASR. This saturation effect suggests that the CSSL pipeline reinforces the backdoor trigger over successive updates, reducing the need for large perturbations.

Notably, the observed stabilization point provides a practical guideline for selecting α. Choosing $\alpha \approx 0.4$ balances attack effectiveness and stealthiness, as it achieves consistently high ASR while limiting the magnitude of injected perturbations. These results demonstrate that the proposed backdoor attack does not rely on excessively large perturbations to remain effective, further underscoring its practicality under realistic black-box constraints.

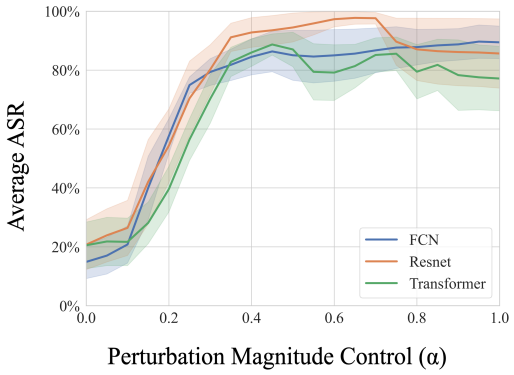


Fig. 4. Effect of perturbation magnitude control α on attack success rate (ASR) for the iAWE dataset.

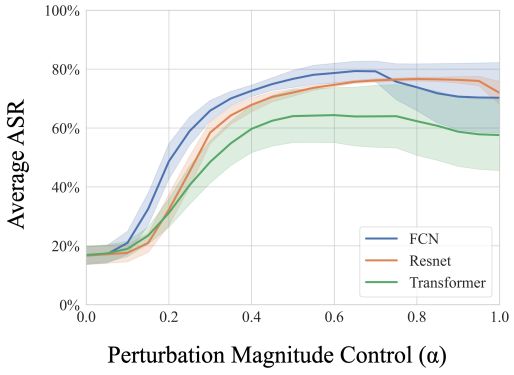


Fig. 5. Effect of perturbation magnitude control α on attack success rate (ASR) for the MotionSense dataset.

VIII. Discussion

Our attack differs fundamentally from conventional image-based backdoor attacks that rely on fixed and input-agnostic trigger patterns (e.g., static patches) [28]. Instead, we adopt a trigger generator that produces input-dependent perturbations, enabling the backdoor to adapt to diverse time-series inputs. As a result, only an adversary with access to the generator can reliably activate the backdoor, which improves flexibility and increases the difficulty of detection in practical deployments.

At the same time, the fully black-box and annotation-free setting introduces intrinsic challenges. Since the attacker cannot manually assign target labels to poisoned samples, a fraction of injected samples may still receive correct pseudo-labels from the CSSL pipeline. These correctly labeled poisoned samples effectively act as adversarial training data and can partially strengthen the model against the backdoor effect [29]. This limitation motivates our surrogate-based optimization strategy, which aims to maximize the likelihood that poisoned samples are pseudo-labeled as the target class, ensuring that they contribute to backdoor reinforcement rather than inadvertently improving model robustness. Nevertheless, this

interaction between pseudo-labeling and backdoor learning highlights an inherent tension in CSSL systems that can be further explored for both attack refinement and defense design.

IX. Conclusion

In this work, we systematically studied backdoor attacks against continual semi-supervised learning (CSSL) in time-series IoT systems under a fully black-box threat model. We proposed the first gradient-free and annotation-free backdoor attack that operates solely through standard query interactions and unlabeled data streams. By combining surrogate model distillation with an input-dependent trigger generator, our approach achieves high targeted attack success rates while inducing only minimal degradation in clean accuracy.

Through extensive experiments and ablation studies, we demonstrated that backdoors implanted under this setting can persist across continual updates and be maintained with infrequent attacker intervention, underscoring a previously unexplored and practical security risk in CSSL-based IoT deployments. Our findings reveal that even highly constrained black-box attackers can exploit the pseudo-labeling and update mechanisms of CSSL to achieve persistent and stealthy backdoor injection.

Future work includes developing robust defenses tailored to continual and semi-supervised learning pipelines, extending the proposed attack framework to other data modalities such as multimodal or multivariate IoT streams, investigating more stealthy trigger designs that can evade advanced detection and mitigation mechanisms, and evaluating the attack under practical defenses such as confidence thresholding and anomaly detection.

References

[1] M. A. Morid, O. R. L. Sheng, and J. Dunbar, "Time series prediction using deep learning methods in healthcare," *ACM Transactions on Management Information Systems*, vol. 14, no. 1, pp. 1–29, Jan. 2023.

[2] V. Dentamaro, V. Gattulli, D. Impedovo, and F. Manca, "Human activity recognition with smartphone-integrated sensors: A survey," *Expert Systems with Applications*, vol. 246, p. 123143, Jan. 2024.

[3] T. C. Nguyen, N. V. Pham, H. N. Truong, N. S. Nguyen, S. T. Cao, N. H. T., J. A. Q. V. Dang, Ware, and N. A. Nguyen, "Electrical load classification using combinative current-power characteristics in time domain and machine-learning-based nilm techniques," in *Proc. of the 3rd Int. Conf. on Advances in Information and Communication Technology (ICTA)*, vol. 1205, Mar. 2024, pp. 912–921.

[4] J. Jiang, B. Wang, Q. Tang, G. Zhong, X. Tang, and J. J. Rodrigues, "Incremental semi-supervised learning for data streams classification in internet of things," *IEEE Transactions on Network and Service Management*, vol. 22, no. 3, pp. 2489–2501, Feb. 2025.

[5] V. Bharti, A. Kumar, V. Purohit, R. Singh, A. K. Singh, and S. K. Singh, "A label-efficient semi self-supervised learning framework for IoT devices in industrial process," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 2253–2262, Jun. 2023.

[6] V. Shejwalkar, L. Lyu, and A. Houmansadr, "The perils of learning from unlabeled data: Backdoor attacks on semi-supervised learning," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Oct. 2023, pp. 4730–4740.

[7] Y. Wang, D. Du, H. Hu, Z. Liang, and Y. Liu, "TSFool: Crafting highly-imperceptible adversarial time series through multi-objective attack," in *27th European Conf. on Artificial Intelligence 2024 (ECAI 2024)*, vol. 392, Oct. 2024, pp. 1422–1429.

[8] Z. Shen and Y. Li, "Temporal characteristics-based adversarial attacks on time series forecasting," *Expert Systems with Applications*, vol. 264, p. 125950, Mar. 2025.

[9] W. Yang, J. Yuan, X. Wang, and P. Zhao, "TSadv: Black-box adversarial attack on time series with local perturbations," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105218, Sep. 2022.

[10] D. Ding, M. Zhang, F. Feng, Y. Huang, E. Jiang, and M. Yang, "Black-box adversarial attack on time series classification," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 37, no. 6, Jun. 2023, pp. 7358–7368.

[11] F. Yin, Y. Zhang, B. Wu, Y. Feng, J. Zhang, Y. Fan, and Y. Yang, "Generalizable black-box adversarial attack with meta learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 3, pp. 1804–1818, Mar. 2024.

[12] D. Ding, M. Zhang, Y. Huang, X. Pan, F. Feng, E. Jiang, and M. Yang, "Towards backdoor attack on deep learning based time series classification," in *2022 IEEE 38th Int. Conf. on Data Engineering (ICDE)*, May 2022, pp. 1274–1287.

[13] Y. Jiang, X. Ma, S. M. Erfani, and J. Bailey, "Backdoor attacks on time series: A generative approach," in *2023 IEEE Conf. on Secure and Trustworthy Machine Learning (SaTML)*, Feb. 2023, pp. 392–403.

[14] X. Lin, Z. Liu, D. Fu, R. Qiu, and H. Tong, "Backtime: Backdoor attacks on multivariate time series forecasting," *Proc. of the 38th Int. Conf. on Neural Information Processing Systems (NeurIPS 2024)*, vol. 37, pp. 131 344–131 368, Dec. 2024.

[15] Y. Huang, M. Zhang, Z. Wang, W. Li, and M. Yang, "Revisiting backdoor attacks on time series classification in the frequency domain," in *Proc. of the ACM on Web Conf. 2025 (WWW '25)*, Apr. 2025, pp. 1795–1810.

[16] C. Dong, Z. Sun, G. Bai, S. Piao, W. Chen, and W. E. Zhang, "TrojanTime: Backdoor attacks on time series classification," in *Proc. of the 29th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, Jun. 2025, pp. 154–166.

[17] S. Koffas, J. Xu, M. Conti, and S. Picek, "Can you hear it? Backdoor attacks via ultrasonic triggers," in *Proc. of the 2022 ACM Workshop on Wireless Security and Machine Learning*, May 2022, pp. 57–62.

[18] T. Zhao, J. Zhang, S. Mao, and X. Wang, "Explanation-guided backdoor attacks against model-agnostic RF fingerprinting systems," *IEEE Transactions on Mobile Computing*, vol. 24, no. 3, pp. 2029–2042, Aug. 2025.

[19] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy (IEEE SP)*, May 2019, pp. 707–723.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd Int. Conf. on Learning Representations 2015 (ICLR 2015)*, May 2015.

[21] I. F. Hassan, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019.

[22] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern recognition letters*, vol. 42, pp. 11–24, Jun. 2014.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, vol. 30, pp. 5998–6008, Dec. 2017.

[24] H. Wang, C.-C. Chang, C.-S. Lu, C. Leckie, and I. Echizen, "Greedypixel: Fine-grained black-box adversarial attack via greedy algorithm," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 12 080–12 095, Nov. 2025.

[25] N. Batra, M. Gulati, A. Singh, and M. B. Srivastava, "It's Different: Insights into home energy consumption in india," in *Proc. of the 5th ACM Workshop on Embedded Systems for Energy-Efficient Buildings (BuildSys 13)*, Nov. 2013, pp. 1–8.

[26] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," in *Proc. of the Int. Conf. on Internet of Things Design and Implementation (IoTDI 19)*, Apr. 2019, pp. 49–58.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778.

[28] Y. Liu, S. Ma, Y. Aafer, W. C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*, Feb. 2018.

[29] F. Tramèr, D. Boneh, A. Kurakin, I. Goodfellow, N. Papernot, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *6th Int. Conf. on Learning Representations (ICLR 2018)*, May 2018.