FEVA-ICS: Benchmarking Adversarial Robustness of Machine Learning-based Intrusion Detection Systems in Industrial Control Systems

Madhurima Ghosh
madhurima.ghosh@cispa.de
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany

Ankush Meshram
ankush.meshram@kit.edu
KASTEL Security Research Labs
Vision and Fusion Laboratory (IES), Karlsruhe Institute of Technology
Karlsruhe, Germany

Markus Karch
markus.karch@iosb.fraunhofer.de
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Karlsruhe, Germany

Christian Haas
christian.haas@iosb.fraunhofer.de
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Karlsruhe, Germany

Xiao Zhang
xiao.zhang@cispa.de
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany

Mridula Singh
singh@cispa.de
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany

Abstract

Machine Learning (ML)–based Intrusion Detection Systems (IDS) are increasingly proposed for deployment in Industrial Control Systems (ICS) to detect evolving and previously unseen attacks. However, ML models are vulnerable to adversarial examples, i.e., carefully crafted inputs that induce misclassification while remaining functionally valid and physically plausible. In safety-critical ICS environments, this vulnerability makes systematic robustness benchmarking essential prior to deployment.

In this paper, we introduce the *Framework for Evasion and Validation for Industrial Control Systems (FEVA-ICS)*, a novel end-to-end benchmarking platform designed to assess ML-based IDS robustness in a realistic black-box setting. *FEVA-ICS* incorporates two attack strategies: (a) a query-based approach and (b) a surrogate model-based approach. In particular, we propose *Correlation-Driven Feature Shift (CorrShift)*, a novel query-based adversarial attack tailored for ICS that preserves physical plausibility and temporal consistency. We also include surrogate-model transfer attacks using gradient-based methods, such as Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD).

Through comprehensive experiments, we show that *CorrShift* consistently outperforms surrogate-based attacks in effectiveness and generalizability, highlighting the importance of ICS-aware adversarial design. The results underscore the need for adversarial robustness evaluation in ML-based IDS pipelines. *FEVA-ICS* establishes a practical and extensible benchmark for adversarial robustness assessment, supporting safer and more reliable deployment of ML-based IDS in real-world ICS environments.

CPSS '26, Bangalore, India

ACM ISBN 979-8-4007-2313-1/26/06
https://doi.org/10.1145/3775042.3807884

CCS Concepts

• **Security and privacy → Intrusion detection systems**; • **Computer systems organization → Embedded and cyber-physical systems**; • **Computing methodologies → Machine learning**.

Keywords

Benchmarking, Adversarial Machine Learning, Intrusion Detection Systems, Industrial Control Systems, Evasion Attacks

ACM Reference Format:
Madhurima Ghosh, Ankush Meshram, Markus Karch, Christian Haas, Xiao Zhang, and Mridula Singh. 2026. FEVA-ICS: Benchmarking Adversarial Robustness of Machine Learning-based Intrusion Detection Systems in Industrial Control Systems. In *12th ACM Cyber-Physical System Security Workshop (CPSS '26), June 1–5, 2026, Bangalore, India.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3775042.3807884

1 Introduction

Industrial Control Systems (ICS) are foundational to the operation of critical infrastructure such as power grids, water treatment facilities, oil and gas pipelines, and manufacturing plants [15, 52]. These systems tightly integrate hardware, software, and physical processes, making them essential for ensuring safety, reliability, and availability across national infrastructure. The increasing connectivity of ICS to corporate networks and the internet has expanded their attack surface significantly. Recent high-profile cyber incidents, such as the Ukraine power grid attack [43], have demonstrated that attacks on ICS can lead to not only data breaches but also widespread physical disruption. As a result, Intrusion Detection Systems (IDS) [27] and anomaly detectors [4] have become a critical component in ICS cybersecurity strategies. In recent year, Machine Learning (ML)-based IDS [31] are being explored as alternative to traditional rule-based IDS. Their inherent characteristic of learning patterns directly from data makes them adaptable to new threats and potentially uncover previously unseen attack vectors. However, despite this promise, the adversarial robustness of ML-based IDS

in ICS remains insufficiently studied. The field of Adversarial Machine Learning (AML) has shown that ML models are susceptible to carefully crafted inputs—known as *adversarial examples*—that can cause misclassifications while remaining inconspicuous to human observers [53]. In ICS contexts, such evasion attacks could allow malicious commands or manipulated sensor readings to bypass detection, possibly leading to dangerous physical consequences.

A closer examination of existing literature reveals several key gaps that hinder the robust and practical adoption of ML-based IDS in real-world ICS deployments. **(1)** Although interest in adversarial robustness is increasing, the literature remain fragmented and largely rely on generic image-domain evasion techniques [3, 9, 17] that overlook ICS-specific structural and operational constraints. While some works incorporate physical constraints [13, 32], temporal modeling is rarely addressed, and evaluations often assume narrow scenarios, such as the targeted model type or the features used for prediction are known.; **(2)** Often studies operate under unrealistic threat models assuming full visibility into model internals and access to the data preprocessing and post-processing pipelines [9, 28, 55, 57], rarely valid in real-world ICS deployments. The preprocessing steps are frequently under-reported and poorly documented, limiting the reproducibility and obscuring their impact on the model's robustness performance.; **(3)** Most adversarial attack strategies in ICS further target IDS models operating on single time steps [3] or non-overlapping windows [21, 57], excluding models that leverage sequential dependencies and restricts insight on how temporal modeling impacts robustness.; and **(4)** The lack of a unified ICS-specific benchmarking framework hinders systematic comparison and realistic evaluation of robustness under consistent and realistic conditions. These gaps motivate two fundamental questions: ***(A)*** *Are ML-based IDS mature and robust enough to replace traditional rule-based systems in real-world ICS environments?* ***(B)*** *If so, what types of robustness evaluations are necessary to enable such deployment?*

To address the aforementioned gaps, this paper investigates following research questions:

RQ1: *How does the robustness of different ML-based IDS architectures vary under the influence of various adversarial attacks?*

RQ2: *How do ICS-specific constraints—such as temporal dependencies, feature correlations and physical laws—affect the feasibility and impact of adversarial attacks?*

RQ3: *Are adversarial examples transferable across different IDS models, and what does this imply for model diversity and defense strategies?*

Through our solutions, we make three key contributions:

(1) We develop *Framework for Evasion and Validation of ICS (FEVA-ICS)*, an end-to-end framework that systematically benchmarks the adversarial robustness of ML-based IDS in ICS environments. *FEVA-ICS* operates without requiring access to the targeted model's architecture or data preprocessing steps, making it suitable for realistic black-box evaluation scenarios. 14 different IDSs are evaluated that cover a diverse range of temporal modeling strategies and ML architectures.
(2) We introduce *Correlation-Driven Feature Shift (CorrShift)*, a novel query-based evasion attack that generates adversarial examples while preserving the temporal and physical constraints inherent in ICS data. In addition, a surrogate model-based attack pipeline approximates black-box models and generates transferable white-box attacks tailored to ICS-specific constraints.
(3) We evaluate the cross-model transferability of adversarial examples to assess how vulnerabilities propagate across different IDS architectures. The impact of variations in temporal modeling strategies is also analyzed, which provides insights into the design of resilient, architecture-agnostic defenses.

Our work aims to provide the ICS Cybersecurity Community valuable insights into the limitations of current ML-based IDS and presents practical evaluation tools to support the development of robust, resilient, and deployable detection systems. In addition, it contributes to the evaluation of the feasibility of ML-based IDS as alternatives or complements to traditional approaches to secure critical infrastructures.

2 Background and Related Work

2.1 ICS

An ICS operates through a hierarchical architecture [49], which involves several layers of control and communication. At the lowest level, sensors and actuators interact with physical processes, providing measurements and executing control actions. These devices communicate with Programmable Logic Controllers (PLCs), which implement real-time control logic. The PLCs then transmit operational data to the Supervisory Control and Data Acquisition (SCADA) system, which offers centralized monitoring, control, and human-machine interfacing. IDS receives data streams from SCADA components to detect anomalous or malicious behaviors in real-time.

2.2 ICS Anomaly Detection Methods

A typical ML-based IDS [42] for ICS consists of three stages: data preprocessing, ML modeling, and post-processing. In the preprocessing stage, raw control process values—typically sensor and actuator data sampled at fixed intervals—are transformed to extract relevant features for anomaly detection. This stage may include time window segmentation [5, 24], statistical feature extraction [48], and digital signal transformations [47, 48] (e.g., Fast Fourier Transform or FIR filters), all aimed at enhancing temporal patterns and reducing noise. Since ICS data is inherently sequential and governed by physical dependencies, these transformations are critical for preserving temporal and contextual information.

Various ML techniques are employed during the modeling stage. Supervised classifiers such as decision trees [56], support vector machines (SVM) [26], and deep neural networks (DNN) [24] are trained on labeled data to distinguish between normal and attack instances. However, because labeled attack data is often scarce in ICS datasets, many approaches rely on unsupervised or semi-supervised learning. Clustering-based outlier detection methods [30, 39] (e.g., k-means, unsupervised k-NN, DBSCAN, Isolation Forests) and reconstruction-based anomaly detectors [14, 45], such as autoencoders and variational autoencoders, are commonly used in such cases. These models are particularly well-suited for identifying zero-day attacks (novel threats not seen during training) by learning

representations of normal behavior and flagging deviations. To capture the temporal dynamics of the system, these models frequently incorporate LSTM or CNN layers that model sequential dependencies and multivariate correlations [44]. The final post-processing stage often includes thresholding softmax probabilities, reconstruction errors, or outlier scores, to classify instances as normal or anomalous [5].

Despite the increasing sophistication of ML-based IDS architectures, much of the existing literature lacks transparency. Many publications do not release their source code, trained models, or even detailed descriptions of their preprocessing pipelines and training data. This makes it difficult to reproduce results, benchmark performance, or assess adversarial robustness across different detection strategies.

2.3 Adversarial Attacks on ICS

In the context of ML-based IDS for ICS, evasion attacks [20] are a class of adversarial strategies that aim to deceive the IDS at inference time. These attacks involve crafting adversarial examples—inputs that are subtly perturbed in a way that causes the IDS to misclassify them, typically labeling malicious behavior as benign and vice versa. The perturbations are designed to be minimal and to preserve the functional and physical plausibility of the input, making them difficult to detect through conventional means.

Existing works have investigated adversarial attacks on ML-based IDS in ICS, primarily focusing on white-box threat models in which the attacker has knowledge of the physical system [34] and full access to the targeted IDS model's architecture, parameters, and gradients [34, 55, 57]. Under such assumptions, gradient-based methods such as Fast Gradient Sign Method (FGSM) [21, 34], Jacob Saliency Map Attack (JSMA) [3], and L_0 Optimization attacks [57] have been widely applied to craft effective adversarial inputs. While these attacks are often effective in experimental settings, they rely on assumptions that rarely hold in real-world ICS deployments.

To address more realistic scenarios, some works adopt a transfer-based attack strategy [3, 9], where a surrogate model is trained to approximate the behavior of the target IDS. Adversarial examples are generated using white-box attacks on the surrogate and then transferred to the black-box target model. Additionally, recent research explores query-based black-box attacks [6], where the attacker has no knowledge of the model internals but can observe outputs in response to crafted inputs. These attacks iteratively probe the model with carefully selected queries to guide perturbations, allowing adversaries to approximate optimal evasion strategies even under strict black-box constraints. Notably, a few recent studies have begun to incorporate physical constraints inherent to ICS—such as valid operational ranges—into their attack generation process [13, 32]. These efforts aim to ensure that adversarial examples remain feasible within the dynamics of real industrial systems and are not trivially detectable through rule-based checks or control logic. The state-of-the-art adversarial attack approaches in ICS, along with their key characteristics, are summarized in Table 1.

In parallel, some defensive strategies have been explored to enhance the robustness of ML-based IDS in ICS. These include adversarial training [3, 9], where models are trained on adversarially perturbed data to improve resilience; robust feature selection [28], which aims to eliminate vulnerable or redundant features; and Approximate Projection Autoencoder (APAE) [21], which integrates approximate projection and feature weighting to significantly improve both model performance and robustness.

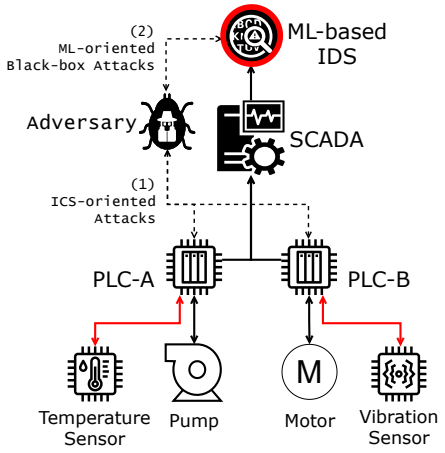

Figure 1: System and threat model: the PLCs control the sensors and actuators through bidirectional communication. SCADA passively monitors process data and feeds it to an ML-based IDS for anomaly detection. An adversary manipulates the sensor data to harm the underlying industrial process (*ICS-oriented Attacks*). Additionally, it has black-box access to the IDS, where only through probing, querying, or reconnaissance, and creates perturbations to misclassify anomalous data as normal (*ML-oriented Black-box Attacks*).

3 Threat Model and Problem Definition

System Model. In this work, we evaluate the adversarial robustness of ML-based IDS deployed in an ICS environment and benchmark their performances. Anomaly detectors that analyze time-series process data from sensors and actuators to detect cyberattacks as anomalies are targeted. We assume an ICS setup with multiple PLCs *(PLC-A/PLC-B)* controlling a physical process through sensors *(temperature/vibration sensors)* and actuators *(motor, pump)*, as shown in Fig. 1. A SCADA system passively monitors network traffic and continuously feeds process data to an ML-based IDS at a fixed time interval. We also assume an adversary is present with intentions to harm industrial processes and evade the detection of malicious events by the IDS, through *ICS-oriented* and *ML-oriented adversarial attacks* respectively. For the attacks on IDS, we consider the '*black-box*' setting where the underlying anomaly detection model is unknown.

3.1 Adversary Capabilities and Assumptions

We assume an adversary with constrained capabilities to alter the industrial process data and gain knowledge about the underlying ML model of the targeted IDS. The detailed constraints in capabilities and knowledge gathering are summarized as follows:

- **Sensor Data Manipulations:** The adversary can perform a man-in-the-middle (MitM) attack on the communication

Table 1: State-of-the-art Evasion Attacks for ML-based ICS-NIDS.

Attack	Access	Targeted Models	Datasets	Comment
GAN-based Attack [16]	BB	AR, LSTM, LDS	Gas Pipeline, SWaT	Substitute model to generate AEs
JSMA [3]	GB	RF, J48	Power Grid	Surrogate models are used
AutoEnc-based Generator Optimisation [13]	WB, BB	NN	BATADAL, WADI	Exploits physical constraints of ICS
L_0 Attacks [57]	WB	LSTM	SWaT	Physical system is known
GAN Attack using Surrogate Model [9]	BB	NN, NB, RF, OC-SVM	CAVS-DA [37]	Incorporates physical constraints.
Best effort Search [32]	WB, GB, BB	LSTM	SWaT	Incorporates physical constraints.
Gradient-based Noise Addition [25]	WB	LSTM	SWaT, WADI	GA to refine perturbations
FGSM, JSMA, DeepFool [17]	WB	NN	CAVS-DA	Compared various WB attacks

Abbreviations: AE=Adversarial Examples; AR=Auto-regressive model; AutoEnc=Autoencoder; BB=Black-box; GA=Genetic Algorithm; GB=Grey-box; LDS=Linear Dynamic State Space; LSTM=Long Short-Term Memory; NN=Neural Network; NB=Naïve Bayes; OC-SVM=One Class-Support Vector Machines; RF=Random Forest; WB=White-box.

channel between sensors and the PLCs. They can perturb sensor readings but cannot directly manipulate actuator commands, reflected by red arrows in Fig. 1.

- **Actuator Imperturbability:** The actuators cannot be directly perturbed by the adversary. This reflects realistic operational constraints in ICS, where the malicious intent of a cyberattack is executed through actuators to cause physical impact. Direct actuator manipulation often requires deep compromise of the PLC or control logic, which may not hold for all adversaries. Moreover, actuators usually operate on binary or discrete values (e.g., ON/OFF states) that are difficult to meaningfully perturb without detection or loss of physical feasibility [46].
- **Limited Knowledge:** The adversary has black-box oracle access to the IDS and can observe its input-output behavior by either actively querying the system or passively gathering information through reconnaissance to see whether an alert is raised. Unlike previous works [3, 13], which assume the adversary knows the type of ML model used for the IDS and its preprocessing pipeline (e.g., the exact window size or feature extraction method), we do not grant this additional knowledge. Instead, we assume a stricter black-box setting in which the attacker must estimate any unknown preprocessing steps—such as the time window size— through probing, querying, or reconnaissance. This makes our threat model more general and realistic for benchmarking.
- **Physical Constraints:** The adversary must respect ICS-specific physical laws and process interdependencies, ensuring that perturbations stay within valid sensor ranges and maintain realistic correlations between components [32]. By correlations, we refer to relationships governed by the control logic of the ICS—for example, an increase in water inflow should logically result in a corresponding rise in tank level, as dictated by the process control rules.
- **Temporal Consistency:** The adversary must ensure that perturbations preserve temporal coherence over time. The perturbation of overlapping data between different time windows must be consistent to reflect real-world time-series behavior. Previous works [28, 34] on AML for ICS often ignore this important temporal constraint. We explicitly enforce it to maintain the realism and feasibility of the generated adversarial examples.

3.2 Adversarial Goals and Approaches

The adversary has two main goals in our black-box setting:

(1) **Conceal Existing Attacks (Evasion):** In order to hide the detection of ongoing *ICS-oriented attacks* by the IDS, craft perturbations to sensor data that cause misclassifications of anomalous behavior as normal. Some prior work [12] refers to this type of attack as *integrity attack*.
(2) **Induce Alert Fatigue:** Generate a high volume of false positives to exhaust the operators in prioritizing and responding to true events. This increases the probability of undetected intrusions and leads to poor detection performance. This objective is sometimes referred to as *availability attack* in previous literature [12].

In order to realize the two goals, we consider two complementary black-box approaches:

(A) Query-based Attacks. The adversary has black-box oracle access and iteratively queries the IDS to refine perturbations without any knowledge of the model architecture, parameters, or gradients. Query-based attacks primarily test the sensitivity of the IDS decision function to small input changes and exploit information leakage through output labels.

(B) Surrogate Model-based Attacks. The attacker trains a surrogate model $f_s(\cdot)$ using a dataset $\mathcal{D}_s = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ that is statistically similar to the targeted IDS's training data $\mathcal{D}_t$ (i.e., $\mathcal{D}_s \approx \mathcal{D}_t$). They use $f_s(\cdot)$ to approximate the targeted model's decision boundaries and craft adversarial examples that transfer to the real IDS.

Query-based attacks assess whether an adversary, without internal knowledge, can induce misclassifications using only output feedback, whereas surrogate model-based attacks evaluate whether

approximating the IDS enables perturbations that remain effective after preprocessing, protocol parsing, or feature extraction.

3.3 Problem Formulation

For the uniform and comprehensive benchmarking of different ML models for robustness, two elements are required: *(1)* access to the process data of an ICS containing normal and anomalies resulting from cyberattacks, and *(2)* different strategies of adversarial attacks against ML models. The open-source ICS datasets such as SWaT and BATADAL align with our system model and partial threat model (*ICS-oriented attacks*), hence they are the evaluation dataset of our benchmarking framework.

Previous works in AML for IDS in ICS have focused on either query-based (Babadi et al. [6]) or surrogate model-based strategies (Chen et al. [9], Li et al. [32]). We include both the approaches to cover the complete threat landscape. This dual strategy ensures that our benchmark reflects the range of realistic black-box adversaries and provides a fair, thorough evaluation across different IDS architectures and deployment scenarios.

At the center of our benchmarking framework is the adversarial example generation, formalized as constrained optimization problem:

$$\min_{\boldsymbol{\delta}} \quad \|\boldsymbol{\delta}\|_p \tag{1}$$

$$\text{s.t.} \quad f(\mathbf{X}+\boldsymbol{\delta}) = \begin{cases} 0, & \text{if } y=1 \quad \text{(evasion)} \\ 1, & \text{if } y=0 \quad \text{(alert fatigue)} \end{cases} \tag{2}$$

$$\forall t,\ \mathbf{x}_t + \boldsymbol{\delta}_t \in \mathcal{P}(\mathbf{x}_t) \tag{3}$$

$$\forall t,\ \|\boldsymbol{\delta}_{t+1} - \boldsymbol{\delta}_t\|_2 < 2\tau \tag{4}$$

$$\forall t,\ \boldsymbol{\delta}_t^{(w)} = \boldsymbol{\delta}_t^{(w')} \quad \text{if } \mathbf{x}_t \in w \cap w'. \tag{5}$$

Here, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ denotes the input time series of length T, $\boldsymbol{\delta} = [\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_T]$ stands for the added perturbations, y represents the ground-truth label (0 for benign, 1 for malicious), and $f(\cdot)$ is the IDS prediction function. The feasible region $\mathcal{P}(\mathbf{x}_t)$ defines valid physical values for each time step. The temporal constraint τ controls how smoothly the perturbation evolves. To ensure temporal realism, overlapping time windows must remain consistent, so if two windows w and w' share a time step $\mathbf{x}_t$, the corresponding perturbations $\boldsymbol{\delta}_t^{(w)}$ and $\boldsymbol{\delta}_t^{(w')}$ must be identical.

In particular, Equation (3) enforces physical feasibility by ensuring that sensor values remain within valid operating ranges after perturbation. While it is desirable also to include correlation-based constraints that reflect physical laws and control logic, we do not model the whole physical system explicitly due to its complexity and the lack of detailed system dynamics under our current assumptions. Instead, we approximate these relationships using patterns learned from the data, which are applied in the query-based attack approach.

The last two constraint equations explicitly enforce temporal consistency and coherence. Equation (4) is incorporated in both black-box approaches by ensuring that the change in perturbation between consecutive time steps is bounded. Specifically, we enforce this constraint by setting the perturbation budget parameter (e.g., ϵ) to be less than τ, allowing controlled perturbations in either direction. Without this constraint, unbounded perturbations could introduce jitters or abrupt changes in sensor values over time, which are unrealistic and violate the physical continuity of industrial processes. For example, a sudden spike in thermometer reading from 10 to 50 units within a single timestep would be physically implausible in most ICS environments and would likely trigger built-in safety mechanisms or manual operator intervention. Thus, maintaining smooth transitions not only preserves attack stealth but also ensures physical plausibility.

Equation (5) is specific to the surrogate model-based approach, where the data is preprocessed into overlapping time windows. To maintain consistency across such windows, we enforce that perturbations to shared time steps remain identical in all windows where they appear. Our black-box approach naturally fulfills this constraint, where we directly perturb the original time series data rather than its preprocessed windowed representation. However, this constraint becomes particularly important in white-box scenarios and surrogate model-based attacks, where overlapping input windows are explicitly constructed and used during training and inference.

4 *FEVA-ICS* Benchmarking Framework

This section introduces the Framework for Evasion and Validation for Industrial Control Systems (FEVA-ICS), a benchmarking framework for systematically evaluating the adversarial robustness of Machine Learning (ML)–based IDS in ICS environments. FEVA-ICS provides comprehensive threat coverage, encompassing both weak and strong adversaries, and evaluates how diverse IDS architectures respond to evasion attacks while respecting the physical constraints and temporal consistency of ICS data. The framework implements a suite of untargeted evasion attacks [41], in which the adversary seeks misclassification without a specific target label. While this study focuses on binary black-box classification of normal and malicious activity, FEVA-ICS also supports white-box robustness evaluation.

4.1 ICS Datasets

Several publicly available datasets have been developed to support research in ICS security, including SWaT [18], WADI [1], BATADAL [50], Gas Pipeline [38], and Power Grid [40] datasets. These datasets simulate different industrial processes and provide labeled data for both normal operation and attack scenarios. Common attack types across these datasets include sensor spoofing, actuator manipulation, command injection, and data integrity violations [10, 36]. These attacks are designed to mimic real-world adversarial behavior, aiming to cause physical damage, disrupt operations, or stealthily alter process outcomes without triggering alarms.

ICS datasets possess several unique characteristics: they consist of heterogeneous multivariate time-series data, involving both discrete and continuous features; they exhibit strong temporal dependencies due to the sequential nature of physical processes; and they are subject to physical and logical constraints, such as mass balance, flow continuity, and control logic safety rules. Furthermore, these datasets are typically highly imbalanced, with attack samples

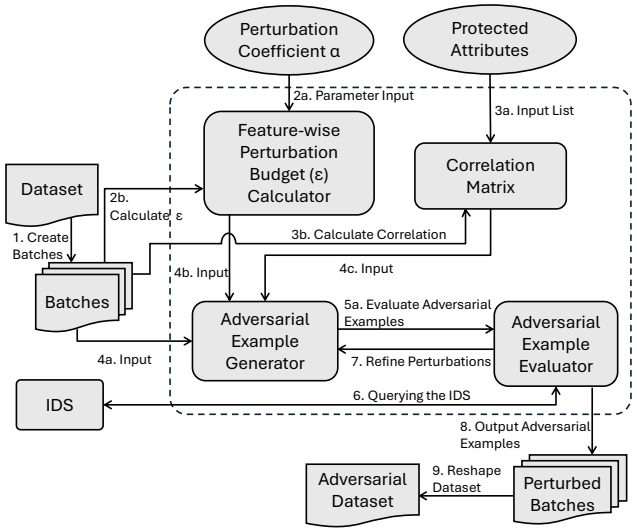

Figure 2: The *CorrShift* attack consists of four main stages: (a) Feature-wise Perturbation Budget Calculation, (b) Correlation Matrix Computation, (c) Adversarial Example Generation, and (d) Adversarial Example Evaluation. The attack processes data in batches and takes as input the batches, perturbation coefficient α, protected attributes, and the targeted IDS. It dynamically computes the feature-wise perturbation budget ϵ and the correlation matrix. Correlation-driven perturbations are then applied to generate adversarial examples. In the final stage, the algorithm evaluates and refines the perturbed samples to maximize misclassification, producing the final adversarial dataset.

constituting a small minority of the data. This class imbalance complicates both detection and adversarial evaluation, as models tend to be biased toward the dominant normal class, often overlooking subtle attack patterns.

In this study, we use the SWaT and BATADAL datasets as representative benchmarks, with their key characteristics summarized in Table 2. The FEVA-ICS framework is, however, dataset-agnostic, allowing users to incorporate additional datasets to evaluate their models as needed.

4.2 Query-based Attack - *CorrShift*

We introduce *Correlation-Driven Feature Shift (CorrShift)*, a novel query-based attack tailored to time-series sensor data in ICS domain. *CorrShift* is specifically designed to preserve the temporal dependencies across time and the interdependencies among sensor components, which are crucial for maintaining the realism of attacks in industrial environments. Unlike prior black-box methods such as the Boundary Attack [7] and Square Attack [2]—designed originally for the image domain and adapted to time series without structural awareness—*CorrShift* explicitly models ICS-specific constraints, including physical validity and temporal consistency. An overview of the attack pipeline is illustrated in Fig. 2. Unlike gradient-based methods, *CorrShift* requires no knowledge of the IDS's internal architecture, gradients, or preprocessing/postprocessing pipeline. Instead, it perturbs input batches and refines them by observing only input-output pairs through black-box queries.

We design *CorrShift* to operate on batches of consecutive time steps, rather than individual inputs. For example, at a 1 Hz sampling rate, each batch may represent a 2-minute window of sensor readings. This strategy enables *CorrShift* to become agnostic to the underlying temporal modeling approach used by the IDS, whether it uses sliding windows, temporal filters, or sequence models such as RNNs.

Perturbing the entire batch with consistent modifications ensures that the temporal smoothness constraint in Equation (4) is naturally satisfied. Specifically, since we add a constant perturbation for each feature across all timestamps, the difference in perturbation between consecutive time steps is always zero, thus remaining well below the threshold τ. Although Equation (5) is not directly relevant to this attack, the design of the *CorrShift* attack—which operates on full batches of data—naturally upholds this constraint by ensuring consistency across any implicit overlapping windows.

Note that ICS datasets are inherently heterogeneous, i.e., each sensor operates over different ranges and scales. A uniform perturbation budget across all features would either over-perturb some or under-perturb others. To address this challenge, we introduce a feature-wise dynamic perturbation budget, defined as:

$$\epsilon_j = \alpha \cdot \left(\max_i X_{i,j} - \min_i X_{i,j} \right), \quad \alpha \in (0,1), \quad \|\delta\|_\infty \leq \epsilon_j, \tag{6}$$

where ϵ_j is the allowable perturbation for feature j, and α is a global scaling coefficient. The perturbation is bounded by the L_∞ norm, ensuring that the maximum absolute change for each feature does not exceed ϵ_j. This keeps the perturbation magnitude proportional to each feature's operating range, maintaining stealthiness and physical feasibility.

To model interdependencies among components, *CorrShift* leverages the idea that statistically correlated features are often functionally linked in an ICS process. For two features i and j, their Pearson correlation coefficient is computed as:

$$\rho_{i,j} = \frac{\mathrm{cov}(X_i, X_j)}{\sigma_{X_i} \sigma_{X_j}}, \tag{7}$$

where $\mathrm{cov}(\cdot)$ is the covariance and σ is the standard deviation. For each batch, *CorrShift* perturbs highly correlated features in the same direction and anti-correlated features in opposite directions. This coupling ensures that the related features shift coherently over time, maintaining the functional plausibility of the adversarial attack.

CorrShift adds a fixed perturbation per feature across all timestamps to preserve smoothness within each batch. After perturbation, we clip each feature's values to remain within the valid physical sensor range (Equation (3)). This step preserves physical plausibility, ensuring that perturbed values are still process-valid and unlikely to trigger safety mechanisms. We also allow users to specify a list of protected features the attacker must not perturb. This is especially important for actuators, which typically carry the malicious intent of a cyberattack by driving physical process changes. It ensures that the *CorrShift* attack only perturbs sensor measurements while respecting the critical separation between sensor spoofing and actuator manipulation.

Table 2: Description of the Datasets used for Experiments.

Features	SWaT [18]	BATADAL [50]
Domain	Water Treatment	Water Distribution
Owner	iTrust Centre	BATADAL Competition Organisers
Testbed	Real	Simulated
Feature Type	Sensors, actuators, network traffic	Sensors, actuators
Data Collection Duration	11 days	1.5 years
Number of Total Datapoints	449920	4178
Number of Attack Datapoints	54584 (~ 12%)	435 (~ 10%)
Number of Normal Datapoints	395336 (~ 88%)	3743 (~ 90%)
Label Type	Binary (Normal = 0, Attack = 1)	Binary (Normal = 0, Attack = 1)
Feature count	51	44
Ground Truth Source	Labeled based on attack timestamps	Roughly Labeled based on attack timestamps
Data Format	XLSX	CSV
Primary Purpose	Anomaly Detection	Benchmarking

The overall attack process is illustrated in Algorithm 1 (in Appendix A). For each sample in the input batch, the *CorrShift* algorithm computes a dynamic perturbation budget ϵ for each feature and identifies non-protected (perturbable) features and computes a correlation matrix among them. For each candidate feature to perturb, the algorithm adjusts all correlated features accordingly—adding or subtracting ϵ based on the sign of their correlation with the candidate. It then evaluates the adversarial impact by querying the targeted IDS model f and selects the perturbation that leads to the maximum number of prediction flips (i.e., misclassifications). The query cost scales linearly with the number of perturbable features, making *CorrShift* efficient for black-box scenarios.

Similar to FGSM, where the direction of the perturbation is guided by the sign of the gradient, *CorrShift* determines the direction of perturbation based on the sign of correlation with the candidate feature. In both cases, the magnitude of the perturbation is not explicitly optimized, leaving room for future work to exploit magnitude-aware strategies for potentially stronger or more stealthy attacks.

4.3 Surrogate Model-based Attacks

In our benchmarking framework, we also implement a surrogate model-based attack pipeline that leverages gradient-based white-box evasion attacks to generate adversarial examples and transfer them to the targeted IDS. This approach unfolds in two major stages: (1) Model Approximation and (2) Adversarial Example Generation, as shown in Fig. 3.

4.3.1 Model Approximation. We first approximate the input dimensions used by the IDS's preprocessing stage — specifically, the window size (in the case of statistical feature extraction or sequence models) or the filter size (in digital signal processing transformations). Our objective is not to recover the exact preprocessing layers or transformation logic. Instead, we only aim to estimate the dimensions of the input data fed into the IDS, which are necessary for maintaining alignment with the temporal overlap constraint (Equation (5)).

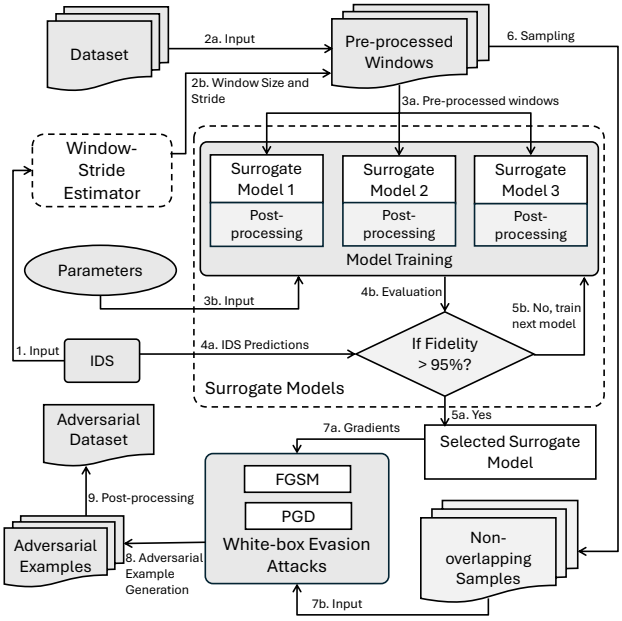

Figure 3: Surrogate model-based attack pipeline. The pipeline consists of two main stages: (a) Model Approximation and (b) White-box Evasion attack. In the approximation stage, the preprocessing characteristics (window size and stride) of the targeted IDS are estimated, and the dataset is transformed accordingly. Surrogate models are then trained to mimic the targeted model's behavior. In the second stage, white-box attacks such as FGSM and PGD are applied using gradients from the surrogate model. The resulting adversarial examples are post-processed to enforce ICS-specific constraints before being evaluated against the targeted IDS.

Recovering the full preprocessing operations would require costly methods such as knowledge distillation [22] or model inversion [51], which demand extensive querying and often rely on assumptions

unavailable in real-world ICS deployments. In contrast, our approach remains lightweight and practical by focusing solely on inferring the length of the processed time window, without reconstructing the actual processing layers.

Assuming the IDS processes fixed-size overlapping windows, we estimate the parameters using [19]:

$$M = \frac{N - L + 2P}{S} + 1, \tag{8}$$

where M is the number of output segments, N is the raw input length, L is the estimated window or filter size, P is padding (if applicable), and S is the stride.

Specifically, we collect input-output pairs through passive observation or active querying. Solving Equation (8) over multiple such input-output pairs yields accurate and consistent estimates of the windowing parameters (window size and stride) because the relationship between input length and output count is deterministic and tightly coupled to the preprocessing configuration. This indirect strategy allows us to align with the IDS's input structure—a crucial step for generating temporally consistent adversarial examples—without requiring insight into internal preprocessing operations. Using the estimated window size and stride, we segment the dataset into overlapping time windows that mimic the targeted IDS's input structure. We then normalize all features to ensure that the later perturbation stage can apply a consistent, feature-agnostic perturbation budget.

Next, we train multiple neural network-based surrogate models on the processed dataset. Each surrogate model learns to approximate the targeted IDS's decision boundaries. The training objective for each surrogate model is defined as:

$$\theta_s^* = \arg\min_{\theta_s} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_s}\Big[\mathcal{L}\big(f_s(\mathbf{x};\,\theta_s),\,y\big)\Big], \tag{9}$$

where θ_s^* denotes the optimal model parameters, f_s is the surrogate model, $\mathcal{L}(\cdot)$ is the classification loss, and $\mathcal{D}_s$ is the shadow dataset used for training. This objective aims to find parameters that minimize the expected prediction error, allowing the surrogate to replicate the decision boundary of the targeted IDS.

In particular, we design three surrogate models with progressively increasing architectural complexity (as detailed in Table 3). All surrogate models output raw logits, which are essential for gradient-based adversarial attacks like FGSM and Projected Gradient Descent (PGD) that depend on loss functions such as logit loss or Carlini-Wagner (C&W) loss. To balance accuracy and computational cost, we adopt a sequential model selection strategy. We start with the simplest model (Surrogate Model 1) and measure its *fidelity*, i.e., the agreement with the IDS's predictions. If it achieves more than 95% fidelity, we skip training the more complex models. If not, we incrementally train Surrogate Models 2 and 3, finally selecting the one with the highest fidelity.

4.3.2 Adversarial Example Generation. In the next stage, we use the gradients of the selected surrogate model to craft transferable adversarial examples. We employ two widely-used white-box evasion attacks: Fast Gradient Sign Method (FGSM) [20] and Projected Gradient Descent (PGD) [35].

While these attacks were originally proposed for general domains such as image classification, we impose an additional constraint when adapting them to the ICS setting: to preserve temporal smoothness as defined in Equation (4), we require that $\epsilon < \tau$, where 2τ denotes the maximum allowable deviation between adjacent timestamps. This constraint was not part of the original attack formulations, but is necessary to ensure the generated perturbations remain realistic and consistent with the dynamics of physical control processes in ICS.

We implement multiple loss functions for PGD, including binary cross-entropy (BCE), logit loss [23], and C&W loss [8], and additionally incorporate Kullback–Leibler (KL) divergence loss [54], which is particularly useful when ground truth labels are unavailable. For FGSM, we approximate an L_0 attack by perturbing only the top-k features with the largest gradients. To preserve temporal consistency (Equation (5)), perturbations are applied only to non-overlapping windows, ensuring coherence across overlapping segments. All adversarial examples are post-processed to maintain functional plausibility by clipping values to valid physical sensor ranges (Equation (3)), reverting actuator states to valid discrete values to preserve malicious intent, and reshaping and rescaling perturbed windows to match the original dataset format.

This surrogate-based approach enables the generation of strong, transferable adversarial examples without extensive probing of the target IDS and remains practical for real-world black-box settings when reliable shadow data is available.

5 Transferability of Adversarial Examples

We evaluated the transferability of adversarial examples across diverse IDSs with varying architectures and preprocessing strategies. Transferability [33] refers to the ability of adversarial examples crafted for a source model f_{src} to cause misclassifications in a different target model f_{tgt}, even without access to its parameters or architecture:

$$f_{\text{src}}(\mathbf{x}) \neq f_{\text{src}}(\mathbf{x}_{adv}) \quad \text{and} \quad f_{\text{tgt}}(\mathbf{x}) \neq f_{\text{tgt}}(\mathbf{x}_{adv}). \tag{10}$$

This property is particularly important in black-box settings, where attackers do not have access to the internal configuration of the IDS but still aim to launch effective attacks. In real-world scenarios, where system internals are often proprietary or protected, transferable adversarial examples pose a serious threat—they can generalize across different deployments, bypassing detection mechanisms even when those systems use different models or data pipelines.

We quantified this phenomenon using the *Transferability Rate* [33], defined as:

$$\textit{Transferability Rate} = \frac{\text{TA}}{\text{N}}, \tag{11}$$

where TA is the number of successful transfer attacks and N is the total number of adversarial examples evaluated. It captures the proportion of attacks that remain effective across models and serves as a strong indicator of generalized vulnerability. High transferability rates suggest that adversarial examples crafted under minimal knowledge assumptions can still bypass diverse IDS defenses—emphasizing the need for architectures that resist such universal attack strategies.

We evaluated not only transfer of adversarial examples from surrogate models to target IDSs, but also direct transfer between

Table 3: Architectures of different Surrogate Models.

Aspect	Surrogate Model 1	Surrogate Model 2	Surrogate Model 3
Architecture	Shallow FNN	Deep FNN	RNN with LSTM layers
Layers	Dense Layers: 128 → 64 → 1	Dense Layers: 256 → 128 → 64 → 32 → 1	LSTM Layers: 64 → 32, Dense Layer
Activations	ReLU for hidden layers	ReLU for hidden layers	ReLU for Dense layer
Regularization	None	L_2 regularization (0.01), dropout (0.3), BN	Dropout (0.3) between LSTM layers
Optimizer	Adam	Adam (with learning rate = 0.01)	Adam (with learning rate = 0.001)
Loss Function	BCE (from logits)	BCE (from logits)	BCE (from logits)

Abbreviations: BN=Batch Normalization; BCE=Binary Cross Entropy; FNN=Feedforward Neural Network; LSTM=Long Short-Term Memory; RNN=Recurrent Neural Network.

Table 4: Description of Targeted IDS.

IDS (#)	ML Model	Data Pre-processing Step	Output Post-processing	Parameters	Dataset
IDS 1	RF [56]	Win (L=4, S=1), **Stat. Feat	None	Scikit-learn Default	SWaT
IDS 2	SVM [11]	Win (L=4, S=1), **Stat. Feat	None	$RBF_{kernel}(\gamma = 0.001)$	SWaT
IDS 3	NB [30]	Win (L=4, S=1), **Stat. Feat	None	Gaussian	SWaT
IDS 4	OC-SVM [24]	Win (L=4, S=1)	0-1 Labeling	$RBF_{kernel}(\nu = 0.0008, \gamma = 0.0046)$	SWaT
IDS 5	DT [56]	Win (L=4, S=1), **Stat. Feat	None	Scikit-learn Default	SWaT
IDS 6	FNN [47]	FIR Filter (cut-off freq.=0.008, taps=101, window=hamming)	0-1 Labeling	Sequential Model (3 layers), AO	SWaT
IDS 7	DT [56]	Win (L=4, S=1), **Stat. Feat	None	Scikit-learn Default	BATADAL
IDS 8	Unsup. k-NN [30]	Win (L=4, S=1), **Stat. Feat	TD (75th%)	Neighbors=10, ED	BATADAL
IDS 9	AAKR [56]	Normalization	TRE (75th%)	Default AAKR	BATADAL
IDS 10	LSTM-AE [45]	Win (L=6, S=1)	TRE (80th%)	LSTM layers, loss=MSE	SWaT
IDS 11	DNN [24]	Win (L=4, S=1)	TOS (85th%)	LSTM layers, loss=MSE	SWaT
IDS 12	CNN [29]	Normalization	0-1 Labeling	1D Convolutional layers	SWaT
IDS 13	NB–AT	Win (L=4, S=1), **Stat. Feat	None	Gaussian	SWaT
IDS 14	DT–AT	Win (L=4, S=1), **Stat. Feat	None	Scikit-learn Default	BATADAL

Abbreviations: AAKR=Auto-Associative Kernel Regression; AO=Adam Optimizer; AT=Adversarial Training; DT=Decision Tree; DNN=Deep Neural Network; ED=Euclidean Distance; FNN=Feedforward Neural Network; Freq.=Frequency; L=Window size; NB=Naïve Bayes; OC-SVM=One Class-Support Vector Machines; RBF=Radial Basis Function; RF=Random Forest; S=Stride; TD=Thresholding Distances; TRE=Thresholding Reconstruction Errors; TOS=Thresholding Outlier Scores; Unsup. k-NN=Unsupervised k-Nearest Neighbors; Win=Windows; **Stat. Feat=Statistical Features(Max, Min, Mean, Standard Deviation).

IDSs with different architectures and preprocessing. In particular, we tested robustness across varying temporal modeling strategies (e.g., window sizes, signal transformations, feedforward vs. LSTM-based models), which are critical for detecting time-dependent anomalies in ICS data. Successful transfer across these variants indicates fundamental weaknesses in learned representations rather than architecture-specific flaws. Overall, this analysis reveals cross-model and cross-preprocessing generalization of adversarial perturbations, exposing systemic vulnerabilities in ICS anomaly detection pipelines.

6 Experimental Setup

In this section, we describe the datasets, IDS models, and attack configurations used in our experiments. We aim to evaluate the robustness of multiple IDSs in realistic ICS scenarios using the proposed *FEVA-ICS*.

Target IDS. We considered a diverse set of IDSs with varying levels of architectural complexity and temporal modeling strategies. Each IDS applied a unique preprocessing pipeline, including time-windowing, sequence modeling (e.g., LSTM), or signal transformations. We list the IDS variants and their architectures in Table 4. This diversity allowed us to examine how attack performance and transferability vary across different IDS characteristics.

Evaluation Configurations. Each IDS was evaluated independently using *FEVA-ICS*, configured with the target model, dataset, and hyperparameters listed in Table 7 (Appendix). For each run, the framework automatically evaluated IDS behavior under normal and

Table 5: Clean & Robustness Performance of Different IDS against CorrShift Attack.

IDS (#)	Clean Acc	Clean Prec	Clean Rec	Rob. Acc	Rob. Prec	Rob. Rec
IDS 1	0.9848	0.9044	0.9784	0	0	0
IDS 2	0.9595	0.8217	0.8511	0.1214	0.1214	1
IDS 3	0.8436	0.4275	0.85	0.5019	0	0
IDS 4	0.9597	0.9703	0.6894	0.124	0	0
IDS 5	0.8468	0.4385	0.9342	0	0	0
IDS 6	0.9776	0.9570	0.8535	0.6735	0.2690	0.9847
IDS 7	0.6828	0.1647	0.5081	0.0024	0	0
IDS 8	0.7799	0.2471	0.5522	0.0382	0.0409	0.3706
IDS 9	0.7017	0.2010	0.6290	0.0010	0	0
IDS 10	0.8158	0.5777	0.8009	0	0	0.0001
IDS 11	0.9141	0.6316	0.7007	0.0062	0	0
IDS 12	0.9971	0.9856	0.9906	0	0	0
IDS 13	0.8436	0.4275	0.85	0	0	0
IDS 14	0.6708	0.1180	0.5916	0.2631	0.0538	0.3706

Abbreviations: Acc=Accuracy; Prec=Precision; Rec=Recall; Rob.=Robustness.

adversarial conditions, recording execution time, classification performance, and robustness metrics. To account for class imbalance and biased decision boundaries common in ICS datasets, separate hyperparameters were used for attack and normal samples.

For fair comparison, identical hyperparameter values were applied across all IDS models. These parameters control the perturbation budget and the physical and temporal constraints enforced during attack generation. While this standardized setup enables consistent benchmarking, we note that model and dataset-specific tuning could further increase attack effectiveness.

FEVA-ICS outputs the *robust accuracy, precision, and recall* of each IDS when exposed to FGSM, PGD, and our proposed *CorrShift* attacks. It also estimates the window size and stride used in preprocessing, evaluates multiple surrogate models, and selects the most effective one based on *fidelity* with the targeted IDS. Additionally, it saves all generated adversarial examples for reproducibility and post-analysis.

Implementation Details. We implemented the framework in Python and ensured full compatibility with NumPy and TensorFlow. We defined each targeted IDS as a Python function that accepts a data matrix and returns binary predictions. We embedded dataset-specific preprocessing and post-processing directly into these functions to maintain modularity and fairness. Each dataset was labeled with ground truth annotations for accurate performance evaluation.

We executed all experiments on a workstation with the following specifications: Intel Xeon CPU @ 3.50GHz, 128 GB RAM, and an NVIDIA GeForce RTX 2080 GPU with 12 GB VRAM, running Ubuntu 24.04.2 LTS. To support reproducibility and community engagement, we have released the full implementation of *FEVA-ICS* on GitHub at https://github.com/Madhurima-Ghosh/FEVA-ICS-Framework.

Table 6: Transferability Property Evaluation of Adversarial Examples (AE) generated using FGSM - BCE and *CorrShift*.

Targeted IDS (#)	AE from IDS (#)	FGSM_{BCE}	*CorrShift*
IDS 1	IDS 11	0.5142	0.9029
IDS 12	IDS 5	0.1214	0.1214
IDS 7	IDS 8	0.6675	0.7369
IDS 8	IDS 7	0.8964	0.9613

7 Results and Analysis

In this section, we evaluate the performance of the proposed *FEVA-ICS*, the attacks included in it, and the transferability of adversarial examples. Due to space constraints, we provide additional experimental results in Appendices D and C, including ablations on the design choice of our *FEVA-ICS* framework.

7.1 Performance of FEVA-ICS

We benchmarked the execution time of evaluating various IDS models using the *FEVA-ICS*. The **IDS 7** (Decision Tree) evaluated on BATADAL has the least execution time of *33.89 seconds*, whereas the **IDS 2** (SVM) evaluated on SWaT dataset took the longest execution time of *2,14,837.39 seconds*, as shown in Table 8 (in Appendix C).

Our results indicate that execution time primarily depends on two factors: (a) the size of the dataset and (b) the complexity of the IDS architecture. Since the SWaT dataset is substantially larger than BATADAL, all IDS models trained and evaluated on SWaT required more time. Additionally, complex models such as **IDS 2** (SVM), **IDS 6** (FNN), **IDS 10** (LSTM Autoencoder), **IDS 11** (DNN), and **IDS 12** (CNN) showed significantly higher execution times. These observations confirm that both data volume and model complexity directly impact benchmarking costs in practical ICS settings.

7.2 Performance of *CorrShift* Attack

We evaluated the impact of the *CorrShift* attack on model robustness by measuring the drop in accuracy, precision, and recall, as shown in Table 5. We observed substantial performance degradation across most IDS models, with several architectures experiencing robust metrics that dropped close to or even reached zero.

Out of the 14 different IDS architectures, only **IDS 2** (SVM) and **IDS 6** (FNN) retained some robustness under *CorrShift*, possibly due to their inherent architectural properties or overfitting tendencies. In particular, the resilience of **IDS 2** may be attributed to overfitting on specific feature patterns, making it less responsive to subtle perturbations. Meanwhile, **IDS 6** applies smoothing filters (FIR Filter) during data preprocessing, which may dampen the effect of feature-wise perturbations and contribute to its relative stability.

These findings validate the strength of the *CorrShift* attack, showing that even without access to model gradients, attackers can craft effective perturbations that preserve the statistical and functional plausibility of ICS data—significantly impairing IDS performance under realistic black-box conditions.

7.3 Comparative Benchmark of Attack Methods

We compared the performance of surrogate model-based and query-based attacks against the baseline performance of IDS models on clean (unperturbed) data. Fig. 4 illustrates the decrease in accuracy, precision, and recall of the IDS models under the influence of adversarial attacks.

Our analysis reveals that in some cases, adversarial examples achieved only the concealment goal (i.e., evasion attacks), while in others, they triggered alert fatigue (i.e., false positives), and in rare cases, both. These patterns suggest that many IDS models exhibit biased decision boundaries—either toward the majority normal class or the minority attack class. Such bias likely results from imbalanced training data, which attackers exploit to launch targeted perturbations that align with these biases. This insight underscores the need to account for data distribution and class balance in IDS training pipelines.

In general, the *CorrShift* attack outperformed the surrogate model-based approach across most IDS architectures. This is primarily because *CorrShift* directly observes model behavior through queries, allowing it to adapt to each model's decision boundary without relying on potentially inaccurate surrogate approximations. Moreover, by explicitly incorporating ICS-specific constraints—such as temporal consistency and sensor correlation—*CorrShift* produces more realistic and stealthy adversarial examples, increasing its effectiveness in black-box settings.

7.4 Transferability of Adversarial Examples

We tested the transferability of adversarial examples generated from *FGSM-BCE* and *CorrShift* attacks on **IDS 11**, **5**, **8**, and **7**, and evaluated their success on **IDS 1**, **12**, **7**, and **8**, respectively. Table 6 presents these results.

CorrShift-based adversarial examples generally achieved higher *Transferability Rate* than those generated using FGSM. This indicates that *CorrShift* generates more generalized and model-agnostic adversarial perturbations, making it particularly dangerous in black-box attack scenarios. These results suggest that attackers can fool multiple IDS models without precisely replicating their architectures. Notably, adversarial examples transferred successfully across diverse detection mechanisms—including classification models and unsupervised learning models—regardless of whether they were gradient-based or not.

However, we also found limits to transferability between models that rely on different temporal modeling strategies. Specifically, adversarial examples crafted for models using sequence-based input (e.g., **IDS 5**) did not effectively transfer to models operating on individual timestamps (e.g., **IDS 12**). This reinforces the importance of accurately estimating the window size and stride used in IDS preprocessing, as detailed in Section 4.3. Understanding these parameters is crucial for designing attacks that maintain effectiveness across heterogeneous IDS architectures.

8 Discussion

In this section, we discuss our findings in the context of how we addressed the research questions raised in the introduction and their security implications for the broader literature of robust IDS development.

Robustness of ML-based IDS architectures (RQ1). Our experiments revealed significant variance in the robustness of different IDS architectures (Fig. 4). Some models, such as SVM (**IDS 2**) and FNN (**IDS 6**), demonstrated limited degradation in performance, while others suffered from severe drops in accuracy, precision, and recall when subjected to adversarial attacks such as FGSM, PGD, and *CorrShift*.

We attribute this variance to differences in decision boundaries, model complexity, and data preprocessing strategies. Several IDS models exhibit biased decision boundaries, often skewed due to class imbalance in training datasets. These biases create exploitable vulnerabilities, allowing adversaries to push samples across class boundaries with minimal perturbation. Additionally, the choice of data preprocessing can significantly influence a model's sensitivity to perturbations and its overall robustness.

Impact of ICS-specific constraints (RQ2). ICS-specific constraints significantly restrict the feasible perturbation space, distinguishing adversarial attack strategies in ICS from those in image or text domains. For instance, perturbing features arbitrarily may violate physical invariants or inter-feature correlations, triggering system-level alarms or operational failures even if an attack is successful against the IDS. In addition, temporal dependencies featured in IDS are enforced through sliding window mechanisms and sequential models, which complicate the attack surface. Overlapping windows imply that a single perturbation may affect multiple samples, but with different gradient directions and magnitudes. This variability undermines the effectiveness of gradient-based methods like FGSM and PGD, and makes gradient approximation techniques in black-box scenarios less reliable in the ICS domain.

Moreover, heterogeneous feature scales in ICS data render the use of a single, global perturbation budget across features ineffective. These findings highlight the need for ICS-specific attack strategies that preserve statistical, temporal, and physical integrity, such as our proposed *CorrShift* method (Subsection 4.2).

Transferability of adversarial examples in ICS context (RQ3). We also investigated whether adversarial examples generated for one IDS can deceive others (Table 6). We observed that adversarial examples often transfer successfully across IDS models with different architectures and learning paradigms. This cross-model transferability poses a substantial threat in black-box scenarios, where attackers may not have access to the targeted IDS but can train a surrogate model to craft transferable perturbations.

However, we also found that transferability breaks down when IDS models differ in their temporal modeling strategies. Specifically, adversarial examples generated for models using sequential inputs often fail to fool models that operate on single-timestamp inputs, and vice versa. This suggests that the ability to conduct successful transferable attacks hinges on accurately estimating the temporal preprocessing parameters of the target IDS, particularly the window size and stride, which shape how input sequences are structured and interpreted.

Implications and Future Work. These findings highlight that increasing model complexity alone is insufficient to ensure adversarial robustness in ML-based IDS. Robust evaluation must instead consider a diverse set of attack strategies, particularly those that respect ICS constraints. Our results indicate that adversarial robustness

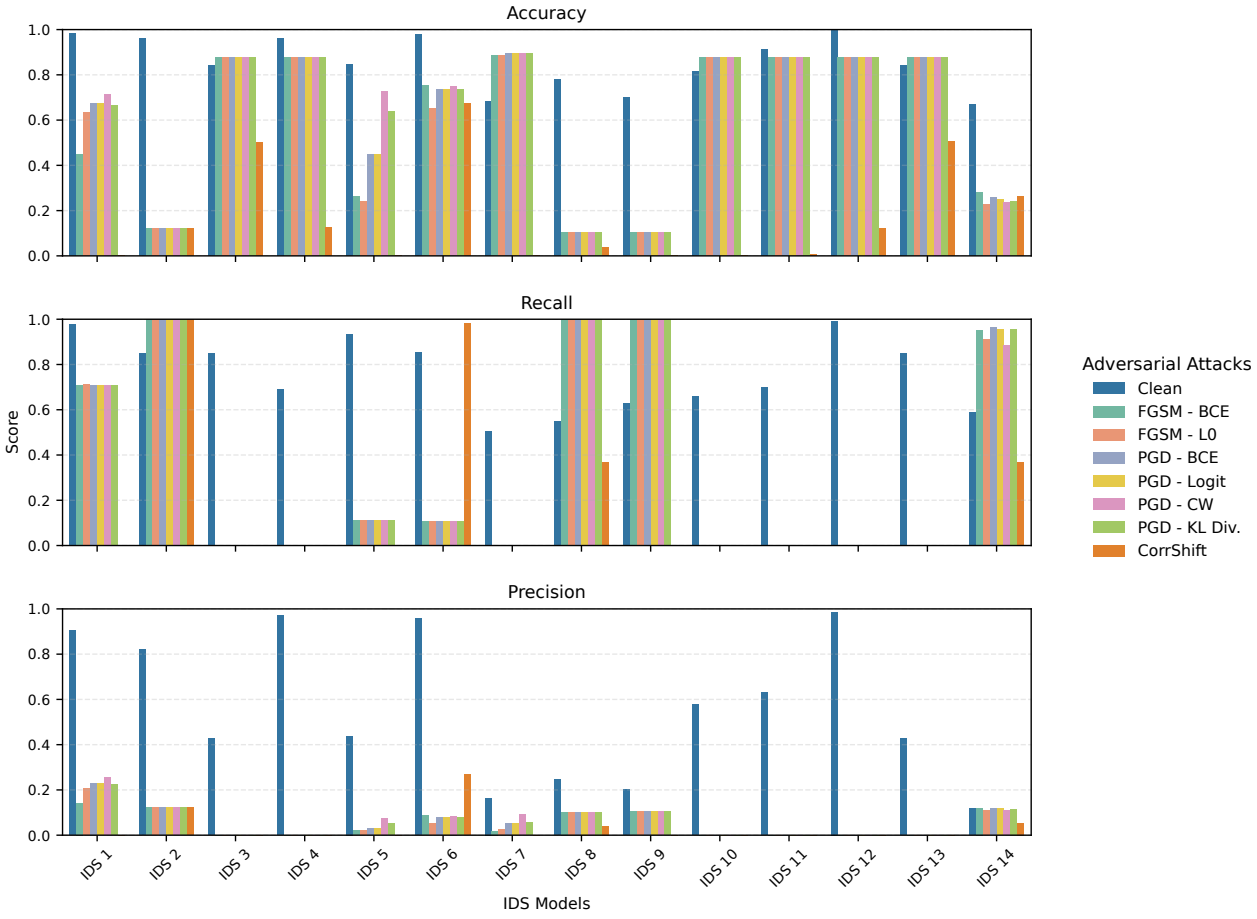

Figure 4: Performance (Accuracy, Recall, and Precision) of IDS models under adversarial attacks. Clean metrics represent the performance of each IDS on unperturbed data. In most cases, we observe that *CorrShift* outperforms other attack methods, demonstrating stronger evasion capability across multiple IDS architectures.

assessment should be a standard component of IDS development pipelines, especially to expose biased decision boundaries arising from data imbalance.

While *FEVA-ICS* enables systematic benchmarking across multiple IDS architectures and attack vectors, our study is limited to publicly available datasets. Consequently, the results may not fully generalize to proprietary ICS environments with distinct system dynamics or access constraints. Future work should therefore explore adversarial attacks that operate without surrogate models or query access.

Finally, our findings suggest that incorporating diverse temporal modeling strategies and robust preprocessing can both improve detection performance and reduce susceptibility to transferable adversarial examples, offering a complementary direction for strengthening IDS robustness.

9 Conclusion

In this paper, we introduced *FEVA-ICS*, a benchmarking framework for evaluating the adversarial robustness of ML-based IDS in ICS. We assessed 14 IDS models using publicly available datasets and two black-box attack strategies, including our novel *CorrShift* attack, which accounts for ICS-specific constraints. Our results show significant robustness variation across IDS architectures and highlight the critical role of temporal dependencies, feature correlations, and physical feasibility in limiting adversarial evasion. While adversarial examples exhibit transferability across models, their effectiveness degrades when temporal modeling strategies differ. Overall, these findings underscore the importance of ICS-aware benchmarking and rigorous robustness evaluation prior to deployment.

Acknowledgments

Partially funded by German Federal Ministry for Digital Affairs and Transport (BMDV) under the project Cypher-AV (Funding Code: 45AVF5A011). Parts of this project were supported by the Topic Engineering Secure Systems of Helmholtz Association.

References

[1] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks* (Pittsburgh, Pennsylvania) *(CySWATER '17)*. Association for Computing Machinery, New York, NY, USA, 25–28. doi:10.1145/3055366.3055375

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*. Springer, 484–501.

[3] Eirini Anthi, Lowri Williams, Matilda Rhode, Pete Burnap, and Adam Wedgbury. 2021. Adversarial attacks on machine learning cybersecurity defences in industrial control systems. *Journal of Information Security and Applications* 58 (2021), 102717.

[4] Muhammad Muzamil Aslam, Ali Tufail, Liyanage Chandratilak De Silva, and Rosyzie Anna Awg Haji Mohd Apong. 2025. Multi-Feature Hybrid Anomaly Detection in ICS: An Integration of ML, DL, and Statistical Techniques. In *Proceedings of the 3rd ACM Workshop on Secure and Trustworthy Deep Learning Systems (SecTL '25)*. Association for Computing Machinery, New York, NY, USA, 43–51. doi:10.1145/3709021.3737669

[5] Muhammad Muzamil Aslam, Ali Tufail, Liyanage Chandratilak De Silva, Rosyzie Anna Awg Haji Mohd Apong, and Abdallah Namoun. 2024. An improved autoencoder-based approach for anomaly detection in industrial control systems. *Systems Science & Control Engineering* 12, 1 (2024), 2334303.

[6] Narges Babadi, Hadis Karimipour, and Anik Islam. 2023. An Ensemble Learning to Detect Decision-Based Adversarial Attacks in Industrial Control Systems. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. 879–884. doi:10.1109/SSCI52147.2023.10371863

[7] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248* (2017).

[8] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. Ieee, 39–57.

[9] Jiming Chen, Xiangshan Gao, Ruilong Deng, Yang He, Chongrong Fang, and Peng Cheng. 2020. Generating Adversarial Examples Against Machine Learning-Based Intrusion Detector in Industrial Control Systems. *IEEE Transactions on Dependable and Secure Computing* PP (11 2020), 1–1. doi:10.1109/TDSC.2020.3037500

[10] Mauro Conti, Denis Donadel, and Federico Turrin. 2021. A survey on industrial control system testbeds and datasets for security research. *IEEE Communications Surveys & Tutorials* 23, 4 (2021), 2248–2294.

[11] Ibrahim Elgendi, Md Farhad Hossain, Abbas Jamalipour, and Kumudu S Munasinghe. 2019. Protecting cyber physical systems using a learned MAPE-K model. *IEEE Access* 7 (2019), 90954–90963.

[12] Alessandro Erba, Riccardo Taormina, Stefano Galelli, Marcello Pogliani, Michele Carminati, Stefano Zanero, and Nils Ole Tippenhauer. 2019. Real-time Evasion Attacks with Physical Constraints on Deep Learning-based Anomaly Detectors in Industrial Control Systems. *ArXiv* abs/1907.07487 (2019). https://api.semanticscholar.org/CorpusID:197430645

[13] Alessandro Erba, Riccardo Taormina, Stefano Galelli, Marcello Pogliani, Michele Carminati, Stefano Zanero, and Nils Ole Tippenhauer. 2020. Constrained concealment attacks against reconstruction-based anomaly detectors in industrial control systems. In *Proceedings of the 36th Annual Computer Security Applications Conference*. 480–495.

[14] Daniel Fährmann, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. 2022. Lightweight long short-term memory variational auto-encoder for multivariate time series anomaly detection in industrial control systems. *Sensors* 22, 8 (2022), 2886.

[15] Xiaohe Fan, Kefeng Fan, Yong Wang, and Ruikang Zhou. 2015. Overview of cyber-security of industrial control system. In *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. 1–7. doi:10.1109/SSIC.2015.7245324

[16] Cheng Feng, Tingting Li, Zhanxing Zhu, and Deeph Chana. 2017. A deep learning-based framework for conducting stealthy attacks in industrial control systems. *arXiv preprint arXiv:1709.06397* (2017).

[17] Henry Figueroa, Yi Wang, and George C Giakos. 2022. Adversarial attacks in industrial control cyber physical systems. In *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 1–6.

[18] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. 2017. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*. Springer, 88–99.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[21] Adam Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. 2021. Robustness of autoencoders for anomaly detection under adversarial impact. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 1244–1250.

[22] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International journal of computer vision* 129, 6 (2021), 1789–1819.

[23] Matthias Hein and Maksym Andriushchenko. 2017. Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation. In *Advances in Neural Information Processing Systems*, Vol. 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/f2d2863674d02c5cb8d1975c5d17b482-Paper.pdf

[24] Jun Inoue, Yoriyuki Yamagata, Yuqi Chen, Christopher M Poskitt, and Jun Sun. 2017. Anomaly detection for a water treatment system using unsupervised machine learning. In *2017 IEEE international conference on data mining workshops (ICDMW)*. IEEE, 1058–1065.

[25] Yifan Jia, Jingyi Wang, Christopher M Poskitt, Sudipta Chattopadhyay, Jun Sun, and Yuqi Chen. 2021. Adversarial attacks and mitigation for anomaly detectors of cyber-physical systems. *International Journal of Critical Infrastructure Protection* 34 (2021), 100452.

[26] Khurum Nazir Junejo and David Yau. 2016. Data driven physical modelling for intrusion detection in cyber physical systems. In *Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016*. IOS Press, 43–57.

[27] Mohamad Kaouk, Jean-Marie Flaus, Marie-Laure Potet, and Roland Groz. 2019. A Review of Intrusion Detection Systems for Industrial Control Systems. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. 1699–1704. doi:10.1109/CoDIT.2019.8820602

[28] Moshe Kravchik and Asaf Shabtai. 2021. Efficient cyber attack detection in industrial control systems using lightweight neural networks and pca. *IEEE transactions on dependable and secure computing* 19, 4 (2021), 2179–2197.

[29] Moshe Kravchik and Asaf Shabtai. 2021. Efficient cyber attack detection in industrial control systems using lightweight neural networks and pca. *IEEE transactions on dependable and secure computing* 19, 4 (2021), 2179–2197.

[30] Philipp Kreimel, Oliver Eigner, and Paul Tavolato. 2017. Anomaly-based detection and classification of attacks in cyber-physical systems. In *Proceedings of the 12th international conference on availability, reliability and security*. 1–6.

[31] Avinash Kumar and Jairo A Gutierrez. 2025. Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure. *Information* 16, 7 (2025), 515.

[32] Jiangnan Li, Yingyuan Yang, Jinyuan Stella Sun, Kevin Tomsovic, and Hairong Qi. 2021. ConAML: Constrained Adversarial Machine Learning for Cyber-Physical Systems. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (Virtual Event, Hong Kong) *(ASIA CCS '21)*. Association for Computing Machinery, New York, NY, USA, 52–66. doi:10.1145/3433210.3437513

[33] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770* (2016).

[34] Yaru Liu, Lijuan Xu, Shumian Yang, Dawei Zhao, and Xin Li. 2024. Adversarial sample attacks and defenses based on LSTM-ED in industrial control systems. *Computers & Security* 140 (2024), 103750.

[35] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[36] Aditya P. Mathur and Nils Ole Tippenhauer. 2016. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. 31–36. doi:10.1109/CySWater.2016.7469060

[37] Thomas Morris, Anurag Srivastava, Bradley Reaves, Wei Gao, Kalyan Pavurapu, and Ram Reddi. 2011. A control system testbed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection* 4, 2 (2011), 88–103. doi:10.1016/j.ijcip.2011.06.005

[38] Thomas Morris, Anurag Srivastava, Bradley Reaves, Wei Gao, Kalyan Pavurapu, and Ram Reddi. 2011. A control system testbed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection* 4, 2 (2011), 88–103. doi:10.1016/j.ijcip.2011.06.005

[39] Sinil Mubarak, Mohamed Hadi Habaebi, Md Rafiqul Islam, Farah Diyana Abdul Rahman, and Mohammad Tahir. 2021. Anomaly Detection in ICS Datasets with Machine Learning Algorithms. *Computer Systems Science and Engineering* 37, 1 (2021), 33–46. doi:10.32604/csse.2021.014384

[40] Shengyi Pan, Thomas Morris, and Uttam Adhikari. 2015. Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems. *IEEE Transactions on Smart Grid* 6, 6 (2015), 3104–3113. doi:10.1109/TSG.2015.2409775

[41] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 372–387.

[42] Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, and Félix J. García Clemente. 2020. MADICS: A Methodology for Anomaly Detection in Industrial Control Systems. *Symmetry* 12, 10 (2020). doi:10.3390/sym12101583

[43] Vetrivel Subramaniam Rajkumar, Alexandru Ştefanov, Alfan Presekal, Peter Palensky, and José Luis Rueda Torres. 2023. Cyber attacks on power grids: Causes and propagation of cascading failures. *IEEE Access* (2023).
[44] Yakub Kayode Saheed, Sanjay Misra, and Sabarathinam Chockalingam. 2023. Autoencoder via DCNN and LSTM Models for Intrusion Detection in Industrial Control Systems of Critical Infrastructures. In *2023 IEEE/ACM 4th International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS)*. 9–16. doi:10.1109/EnCyCriS59249.2023.00006
[45] Peter Schneider and Konstantin Böttinger. 2018. High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy* (Toronto, Canada) *(CPS-SPC '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3264888.3264890
[46] Ryan Sheatsley, Nicolas Papernot, Michael J. Weisman, Gunjan Verma, and Patrick McDaniel. 2022. Adversarial examples for network intrusion detection systems. *J. Comput. Secur.* 30, 5 (Jan. 2022), 727–752. doi:10.3233/JCS-210094
[47] Alexander N Sokolov, Andrey N Ragozin, Ilya A Pyatnitsky, and Sergei K Alabugin. 2019. Applying of digital signal processing techniques to improve the performance of machine learning-based cyber attack detection in industrial control system. In *Proceedings of the 12th International Conference on Security of Information and Networks*. 1–4.
[48] Melissa Stockman, Dipankar Dwivedi, Reinhard Gentz, and Sean Peisert. 2019. Detecting control system misbehavior by fingerprinting programmable logic controller functionality. *International Journal of Critical Infrastructure Protection* 26 (2019), 100306.
[49] K. Stouffer, J. Falco, and K. Scarfone. 2011. Guide to Industrial Control Systems (ICS) Security. https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final. Accessed: 2025-03-23.
[50] Riccardo Taormina, Stefano Galelli, Nils Ole Tippenhauer, Elad Salomons, Avi Ostfeld, Demetrios G. Eliades, Mohsen Aghashahi, Raanju Sundararajan, Mohsen Pourahmadi, M. Katherine Banks, B. M. Brentan, Enrique Campbell, G. Lima, D. Manzi, D. Ayala-Cabrera, M. Herrera, I. Montalvo, J. Izquierdo, E. Luvizotto, Sarin E. Chandy, Amin Rasekh, Zachary A. Barker, Bruce Campbell, M. Ehsan Shafiee, Marcio Giacomoni, Nikolaos Gatsis, Ahmad Taha, Ahmed A. Abokifa, Kelsey Haddad, Cynthia S. Lo, Pratim Biswas, M. Fayzul K. Pasha, Bijay Kc, Saravanakumar Lakshmanan Somasundaram, Mashor Housh, and Ziv Ohar. 2018. Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. *Journal of Water Resources Planning and Management* 144, 8 (2018), 04018048. doi:10.1061/(ASCE)WR.1943-5452.0000969
[51] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*. 601–618.
[52] Muhammad Azmi Umer, Khurum Nazir Junejo, Muhammad Taha Jilani, and Aditya P Mathur. 2022. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection* 38 (2022), 100516.
[53] Apostol Vassilev, Alina Oprea, Alie Fordyce, and Hyrum Andersen. 2024. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. doi:10.6028/NIST.AI.100-2e2023
[54] Yisen Wang, Difan Wang, Yisen Zhang, and Michael I Jordan. 2019. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJlQnkFDS
[55] Lijuan Xu, Zhiang Yao, Dawei Zhao, and Xin Li. 2024. GNN-ASG: A Double Feature Selection-based Adversarial Sample Generation Method in Industrial Control System. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 359–364.
[56] Fan Zhang, Hansaka Angel Dias Edirisinghe Kodituwakku, J Wesley Hines, and Jamie Coble. 2019. Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. *IEEE Transactions on Industrial Informatics* 15, 7 (2019), 4362–4369.
[57] Giulio Zizzo, Chris Hankin, Sergio Maffeis, and Kevin Jones. 2020. Adversarial Attacks on Time-Series Intrusion Detection for Industrial Control Systems. 899–910. doi:10.1109/TrustCom50675.2020.00121

A CorrShift Attack Algorithm

We illustrate the *CorrShift* attack in Algorithm 1.

Algorithm 1 *Correlation-Driven Feature Shift (CorrShift)* Attack

Require: Model f, Input Data X, True label y, Perturbation Coefficient α, List $protected\_columns$
Ensure: Perturbed Data $\tilde{X}$

Initialize $\epsilon \leftarrow None$
$(num\_samples, n_{\text{rows}}, n_{\text{cols}}) \leftarrow \text{shape}(X)$
for $j \leftarrow 1$ to n_{cols} **do**
  Dynamically calculate Perturbation Budget ϵ for each feature: $\epsilon[j] \leftarrow \alpha \cdot (\max_{i=1}^{n_{\text{rows}}} X_{i,j} - \min_{i=1}^{n_{\text{rows}}} X_{i,j})$
end for
Compute Correlation Matrix $corr\_matrix$ for $valid\_attributes$
Initialize $max\_diff \leftarrow -1$
Initialize $best\_adv\_examples \leftarrow None$
Initialize $\tilde{X} \leftarrow X$
Identify non-protected columns: $valid\_attributes \leftarrow \{i \mid i \in [0, num\_features), i \notin protected\_columns\}$
for each $sample\_idx$ in $num\_samples$ **do**
  $original\_sample \leftarrow X[sample\_idx]$
  for each $candidate\_feature$ in $valid\_attributes$ **do**
    $adv\_example \leftarrow original\_sample$
    for each $feature$ in $valid\_attributes$ **do**
      if $feature = candidate\_feature$ **then**
        $adv\_example[:, feature] \leftarrow adv\_example[:, feature] + \epsilon[feature]$
      end if
      if $corr\_matrix[candidate\_feature, feature] < 0$ **then**
        $adv\_example[:, feature] \leftarrow adv\_example[:, feature] - \epsilon[feature]$
      else
        $adv\_example[:, feature] \leftarrow adv\_example[:, feature] + \epsilon[feature]$
      end if
    end for
    $predictions \leftarrow f(adv\_example)$
    Compute $diff\_count \leftarrow \sum(predictions \neq y[sample\_idx])$
    if $diff\_count > max\_diff$ **then**
      Update $max\_diff \leftarrow diff\_count$
      Update $best\_adv\_examples \leftarrow adv\_example$
    end if
  end for
  $\tilde{X}[sample\_idx] \leftarrow best\_adv\_examples$
end for
return $\tilde{X}$

B Hyperparameters for *FEVA-ICS*

Table 7 describes the different hyperparameters for *FEVA-ICS* and values we have set for our evaluation.

C Evaluation

C.1 Execution Time of FEVA-ICS

Table 8 compares the execution time to benchmark the adversarial robustness of different IDS architectures.

C.2 Performance of Surrogate Models

Table 9 summarizes the performance of these surrogate models. In several cases, we train only the first surrogate model, as it achieves a fidelity greater than 95%, eliminating the need to train more complex models and thereby optimizing the process.

Table 7: Hyperparameters of *FEVA-ICS*.

Parameters	Description	SWaT	BATADAL
NUM_FEATURES	Number of Features in the Dataset	51	44
N	Batch Size	60	8
ϵ_{normal}	FGSM/PGD Perturbation Budget for Normal Samples	0.1	0.1
ϵ_{attack}	FGSM/PGD Perturbation Budget for Attack Samples	0.001	0.001
k_{normal}	Perturbed Components in FGSM-L_0 Attack for Normal Samples	50	40
k_{attack}	Perturbed Components in FGSM-L_0 Attack for Attack Samples	25	25
α_{normal}	PGD Attack Step Size for Normal Samples	0.01	0.01
α_{attack}	PGD Attack Step Size for Attack Samples	0.0001	0.0001
$\alpha_{corrshift}$	Perturbation Coefficient for *CorrShift* attack	0.01	0.01
κ	Confidence Parameter of CW loss	0.5	0.5
Protected Columns	List of Imperturbable Features	Binary Actuators	Binary Actuators

Table 8: Execution Time of *FEVA-ICS* for Various IDS Models

IDS (#)	Execution Time (s)
IDS 1	2710.42
IDS 2	214837.39
IDS 3	1771.69
IDS 4	7311.27
IDS 5	3848.59
IDS 6	177659.82
IDS 7	33.89
IDS 8	145.98
IDS 9	819.34
IDS 10	13785.97
IDS 11	36520.69
IDS 12	38154.47
IDS 13	1571.46
IDS 14	35.17

D Ablation Studies

We evaluate the effectiveness of specific design choices within the *FEVA-ICS* framework by conducting ablation experiments targeting two key components of the adversarial example generation process. These experiments focus exclusively on surrogate model-based attacks.

First, we examine the impact of distinct perturbation constraints on normal and attack samples. By removing this distinction and applying a uniform constraint across all samples, we observe changes in both the success rate of attacks and the quality of the adversarial examples. This experiment allows us to quantify the contribution of adaptive perturbation constraints in enhancing the attack's effectiveness while maintaining realism.

As shown in Table 10, the use of distinct perturbation constraints leads to a decrease in *robust accuracy, precision,* and *recall* – indicating more effective attacks. This drop in performance metrics reflects the improved ability of the attack to fool the IDS, especially by preserving realism in the perturbed attack samples. By assigning tighter constraints to attack samples, we minimize noticeable distortions while still successfully altering the IDS decisions.

Second, we assess the role of post-processing by replacing the variance-based resolution mechanism with a simpler sampling of non-overlapping windows. This method resolves overlapping predictions by selecting outputs based on variance, aiming to improve the quality of adversarial sequences. To evaluate its effectiveness, we compare it against a simpler strategy: sampling non-overlapping windows from the dataset. We compare the performance of the generated adversarial examples under both approaches to determine which method provides a measurable benefit in preserving temporal coherence and attack stealthiness.

As shown in Table 11, both methods produce similar results, with metric deviations of less than 1%. This small difference indicates that the added complexity of variance-based resolution does not yield a meaningful improvement in attack effectiveness. Given the minimal performance gap and the higher computational cost of the variance-based approach, we select the sampling method for our final framework. This choice reduces computational overhead while maintaining the quality and impact of the adversarial attacks.

Table 9: Surrogate models performance across different IDS. Metrics include Accuracy (Acc), Precision (Prec), Recall (Rec), and Fidelity (Fid) for all surrogate models - Surrogate Model 1 (SM1), Surrogate Model 2 (SM2) and Surrogate Model 3 (SM3).

Metric	IDS 1	IDS 2	IDS 3	IDS 4	IDS 5	IDS 6	IDS 7	IDS 8	IDS 9	IDS 10	IDS 11	IDS 12	IDS 13	IDS 14
Acc (SM1)	0.967	0.937	0.851	0.911	0.877	0.975	0.679	0.578	0.780	0.800	0.870	0.989	0.840	0.561
Prec (SM1)	0.967	0.932	0.439	0.628	0.674	0.959	0.150	0.143	0.152	0.566	0.761	0.961	0.639	0.135
Rec (SM1)	0.882	0.790	0.845	0.987	0.923	0.839	0.531	0.718	0.507	0.632	0.648	0.953	0.734	0.697
Fid (SM1)	**0.983**	**0.993**	**0.996**	**0.998**	0.852	**0.994**	0.743	**0.903**	**0.839**	**0.990**	**0.983**	**0.991**	**0.998**	0.725
Acc (SM2)	-	-	-	-	0.922	-	0.843	0.473	0.919	-	-	-	-	0.711
Prec (SM2)	-	-	-	-	0.985	-	0.210	0.100	0.352	-	-	-	-	0.155
Rec (SM2)	-	-	-	-	0.680	-	0.250	0.593	0.260	-	-	-	-	0.479
Fid (SM2)	-	-	-	-	0.835	-	0.702	0.744	0.729	-	-	-	-	0.545
Acc (SM3)	-	-	-	-	0.881	-	0.725	0.580	0.821	-	-	-	-	0.569
Prec (SM3)	-	-	-	-	0.678	-	0.179	0.140	0.165	-	-	-	-	0.145
Rec (SM3)	-	-	-	-	0.935	-	0.552	0.687	0.420	-	-	-	-	0.750
Fid (SM3)	-	-	-	-	**0.889**	-	**0.776**	0.887	0.796	-	-	-	-	**0.756**

Table 10: Comparison of white-box attack performance on IDS 1 under two constraint settings: unified and distinct. For the unified setting, we apply $\alpha = 0.01$ and $\epsilon = 0.1$. In contrast, the distinct constraint setting uses separate parameters for normal and attack samples: $\alpha_{\text{normal}} = 0.01$, $\alpha_{\text{attack}} = 0.0001$, $\epsilon_{\text{normal}} = 0.1$, and $\epsilon_{\text{attack}} = 0.001$. In both configurations, we maintain $\kappa = 0.5$.

Attack	Acc(U)	Prec(U)	Rec(U)	Acc(D)	Prec(D)	Rec(D)
FGSM – BCE	0.6107	0.329	0.7592	**0.4475**	**0.1426**	**0.7085**
FGSM – L_0	0.9267	0.9406	0.7076	**0.6339**	**0.2070**	**0.7120**
PGD – BCE	0.7095	0.4120	0.7840	**0.6761**	**0.2298**	**0.7095**
PGD – Logit	0.7093	0.4115	0.7818	**0.6759**	**0.2297**	**0.7094**
PGD – CW	0.6956	0.3969	0.7679	**0.7121**	**0.2544**	**0.7107**
PGD – KL div.	0.6965	0.3999	0.7889	**0.6664**	**0.2242**	**0.7105**

Abbreviations: Acc=Accuracy; D=Distinct; Prec=Precision; Rec=Recall; U=Unified.

Table 11: Comparison of attack effectiveness using variance-based resolution and non-overlapping window sampling on IDS 1. Metrics show less than 1% deviation, supporting the use of the more efficient sampling approach.

Attack	Acc(S)	Prec(S)	Rec(S)	Acc(V)	Prec(V)	Rec(V)
FGSM – BCE	0.4482	**0.1425**	0.7088	**0.4475**	0.1426	**0.7085**
FGSM – L_0	**0.6337**	**0.2070**	**0.7115**	0.6339	**0.2070**	0.7120
PGD – BCE	**0.7095**	**0.4120**	**0.7840**	0.7205	0.4122	0.7848
PGD – Logit	**0.7093**	**0.4115**	0.7818	**0.7093**	**0.4115**	**0.7817**
PGD – CW	**0.6956**	**0.3969**	**0.7679**	0.6957	0.3970	0.7682
PGD – KL div.	0.6965	0.3999	**0.7889**	**0.6962**	**0.3996**	**0.7889**

Abbreviations: Acc=Accuracy; Prec=Precision; Rec=Recall; S=Sampling; V=Variance.