electronics

MDPI

Article

Automated Identification and Interpretation of Anomalous Cases in Industrial Control Systems

Seonwoo Lee [1], Seungbeom Lim [2] and Taejin Lee [1,*]

[1] Department of Smart Security, Gachon University, Seongnam 13120, Republic of Korea; dltjs5621@gachon.ac.kr
[2] SANDS Lab, Gwacheon 13840, Republic of Korea; lsb51490@gmail.com
* Correspondence: kinjecs0@gmail.com

Abstract

Industrial control systems (ICS), which manage critical infrastructure such as power grids and water treatment, are increasingly exposed to cyber threats and operational faults as their connectivity to external networks grows. AI-based anomaly detection has emerged as a key defense, yet three limitations restrict its practical deployment: (i) detected anomalies are treated uniformly without distinguishing between transient faults and intentional attacks, hindering tailored incident response; (ii) the trade-off between detection accuracy and the false-positive rate burdens experts with extensive manual triage and delays prompt action; and (iii) prevailing feature-attribution Explainable AI (XAI) techniques such as SHAP and LIME produce fragmented sensor-level explanations and fail to capture correlations among sensors in time-series data, undermining trust in model decisions. To address these gaps, this paper proposes a graph-based deep learning framework that (a) defines anomaly types in terms of the anomalous-sensor ratio measured before and after smoothing—which operationalizes the correlation-maintenance principle that faults keep coupled sensors jointly anomalous while attacks isolate them—enabling explicit separation of faults, attacks, false positives, and false negatives; (b) identifies ambiguous decisions near the detection threshold as candidate false alarms via dynamic threshold smoothing; and (c) provides correlation-aware graph visualizations for intuitive interpretation. Experiments on the Secure Water Treatment (SWaT) dataset center on this post-detection layer: built on a standard graph-based detector (F1-score 0.787 at Top-K = 10) that serves only as the substrate, the categorization separates faults from attacks, and the subsequent ambiguity analysis identifies false negatives with 83% precision and false positives with 73% precision. By separating attacks from faults and surfacing high-likelihood false alarms together with intuitive sensor-correlation explanations, the proposed approach reduces analyst workload and supports more reliable, prioritized incident response in ICS environments.

Keywords: anomaly detection; graph; correlation; attack; fault; false alarm

Check for updates

Academic Editor: Aryya Gangopadhyay

Received: 16 May 2026
Revised: 11 June 2026
Accepted: 16 June 2026
Published: 18 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Artificial intelligence (AI) has moved beyond convenience-oriented applications into safety- and security-critical domains, including autonomous vehicles, medical diagnostics, and industrial process control [1]. In these settings, AI no longer merely suggests outcomes; it participates in or triggers actions whose failure carries operational or human cost. This expansion places new demands on how AI systems are validated, interpreted, and supervised.

Industrial control systems (ICS), which operate national infrastructure such as electricity, water, and gas networks, have traditionally been isolated and optimized for availability. The push toward smart environments has increasingly connected these systems to external networks, raising their exposure to cyber threats [2,3]. AI-based anomaly detection has emerged as a principal defense against both faults and attacks in ICS [4], but its operational deployment faces three persistent challenges.

The first challenge is that anomaly types are not consistently defined. Faults and attacks are frequently collapsed into a single "anomaly" class, even though they demand very different responses [5]: a transient fault driven by environmental change may self-correct, whereas an attack that deliberately manipulates sensor readings can produce persistent and operationally critical effects. Without a principled way to separate the two at detection time, every flagged event must be triaged uniformly, which is inefficient in availability-critical environments.

The second challenge is the well-known trade-off between detection accuracy and the false-positive rate [6]. Pushing recall higher tends to inflate false positives, lengthening expert review and delaying action. Tools that can flag ambiguous decisions near the detection threshold—rather than simply tuning the threshold itself—would shorten this loop and surface cases worth re-examining.

The third challenge is interpretability. Modern AI models behave as black boxes, and trusting their decisions in ICS requires understanding why a given time point is flagged [7,8]. Mainstream Explainable AI (XAI) techniques applied to time series [9,10] attribute importance to individual sensors but rarely capture the correlations between them, which is precisely the information an expert needs to trace how an anomaly propagates through coupled sensors.

These challenges are concrete in practical ICS operations. In furnace temperature control, where a heater must track a setpoint under wireless IoT supervision [11], a maliciously manipulated temperature reading and a transient thermal disturbance call for opposite responses; likewise, in water-level regulation at treatment plants and in pressure and strain monitoring of natural-gas pipelines [12], a localized sensor fault and a coordinated attack demand different interventions. In each setting the proposed framework helps operators by separating attacks from faults so that high-impact cases are prioritized, by surfacing likely false alarms before they trigger unnecessary shutdowns, and by tracing—through sensor correlations—which downstream sensors a disturbance is likely to reach.

These three challenges are not independent; they all hinge on the same underlying signal—how sensors co-vary around a flagged time point. Whether an event is a fault or an attack, whether a near-threshold decision is a false alarm, and how an anomaly propagates across sensors are all read from a single per-time-point structure: the top-K correlation graph together with the pre- and post-smoothing counts of anomalous sensors. Addressing them as separate tasks would recompute this structure redundantly and risk mutually inconsistent verdicts, whereas deriving all three from one analysis lets categorization, false-alarm identification, and interpretation reinforce one another. We therefore address them jointly within a single post-detection framework.

The main contributions of this paper are summarized as follows:

1. Correlation-based anomaly type definition. We define faults, attacks, false positives, and false negatives in terms of the anomalous-sensor ratio before and after smoothing, which operationalizes the correlation-maintenance principle (faults keep correlated sensors jointly anomalous; attacks isolate them), enabling prioritized response to high-impact attacks rather than uniform triage of every detected anomaly.
2. Ambiguity-aware false alarm identification. We propose a dynamic threshold smoothing procedure that flags decisions near the detection threshold as candidate false

positives or false negatives, thereby reducing analyst workload and surfacing concrete cases that can inform future model refinement.

3. Correlation-aware graph interpretation. We introduce a graph-based interpretation method that complements feature-attribution XAI such as SHAP and LIME by exposing sensor-to-sensor correlations at decision points, providing intuitive causal explanations that are otherwise fragmented in time-series settings.

The paper is structured as follows: Section 2 discusses the challenges in defining anomalies and the reliability of AI model judgments. Section 3 reviews related studies on Graph Neural Networks, anomaly detection, and XAI. Section 4 details the proposed graph-based model and the methods for interpreting faults, attacks, and false positives/negatives. Section 5 presents experimental results using the SWaT dataset, illustrating the application of the proposed methodology. Section 6 discusses the experimental validation results and their contribution to improving anomaly detection. Finally, Section 7 outlines future work and concludes the paper.

2. Motivation

2.1. Lack of Definition for Anomaly Types: Challenges in Responses

Existing anomaly detection systems do not provide a principled way to separate faults from attacks. The term "anomaly" is defined broadly—typically as any deviation from learned normal behavior—and its interpretation varies across studies, with few works anchoring it in specific observable criteria. As a result, every flagged time point must be triaged the same way, regardless of whether it stems from a benign environmental fluctuation or a deliberate intrusion that demands immediate response. Compounded by the accuracy/false-positive trade-off, this blanket treatment imposes excessive expert workload and delays action precisely where availability matters most. A categorization scheme that ties anomaly type to observable characteristics, rather than ad hoc threshold tuning, would allow analysts to prioritize the high-impact cases without scrutinizing every detection.

2.2. AI Model Interpretability and Time-Series XAI Limitations

With the enhancement in the performance of AI models, a concurrent increase in their complexity is observed, which adversely affects their interpretability. This challenge is particularly pronounced in the context of time-series data, where interpretative studies are relatively less developed.

The two most widely adopted post hoc XAI techniques, LIME [7] and SHAP [13,14], compute per-feature attribution scores by perturbing inputs and observing the change in model output. While effective for tabular or single-time-step classification, they exhibit two structural limitations in ICS time-series settings. First, they treat sensors as independent contributors: the attribution score for each sensor is computed in isolation, so coupled sensors that move together—for example, a pressure sensor and a downstream flow sensor sharing a hydraulic link—are reported independently with no signal that they form a single physical chain. Consequently, when an attack on a single sensor cascades through correlated sensors, feature-attribution methods flag all affected sensors with comparable importance, leaving the expert to manually trace which sensor is causal and which is downstream. For instance, in the SWaT testbed used in our experiments, a manipulation of the upstream pH sensor AIT202 propagates anomalies to AIT402 and FIT301 within minutes (Case 3 in Section 5); a feature-attribution method will surface all three sensors but cannot expose the dependency that makes AIT202 the actionable root cause. Second, mainstream SHAP and LIME variants explain a single time step or a flattened window, so persistent correlations between sensors over time are not part of the explanation. A recent survey on

XAI for time-series data [6] confirms this gap: existing feature-attribution methods provide fragmented, time-step-local explanations rather than coherent interpretations across sensor interdependencies.

These limitations together leave a gap that this paper aims to fill: an explanation that explicitly represents sensor-to-sensor correlations and exposes how they change at decision points. By constructing a correlation graph over the top-K key sensors at each time point (Section 4), the proposed approach renders the physical and statistical couplings as edges, so that experts can read off both the focal anomalous sensor and the chain of correlated sensors that warrant follow-up. Correlation heatmaps and impact-sensor subgraphs further translate these dependencies into intuitive visual artifacts, enabling the prompt, prioritized response that availability-critical ICS environments demand.

3. Related Work

A graph neural network (GNN) is a type of machine learning model designed to process data structured in the form of graphs, comprising nodes and edges. This model is particularly adept at handling the complex, interconnected nature of graph-shaped data, common in various real-world scenarios. For instance, chemical compounds can be effectively represented as graphs, with nodes symbolizing atoms or chemical entities and edges denoting chemical bonds. Similarly, the structure of the web can be modeled as a graph, where individual webpages are the nodes and the hyperlinks between them constitute the edges [15]. The primary function of a GNN is to model the interactions between each node and its adjacent nodes. Utilizing neural networks, GNNs learn to create embeddings that encapsulate the characteristics of each node. These learned embeddings can then be applied to predict or analyze interactions among the nodes.

The foundational algorithms of GNN models include message passing and readout [16]. Message passing involves each node in the graph receiving and integrating information from its neighboring nodes, thereby updating its state. During this process, nodes assimilate messages from their connected neighbors and amalgamate these messages to revise their state. The readout process aggregates information across all nodes to deduce the overall characteristics or representation of the entire graph. To refine this aggregation, methods such as the graph attention network (GAT) have been introduced, which allocate weights based on the importance of information, rather than treating all nodes equally [17]. A parallel line of work studies when such neighborhood aggregation helps, showing that its benefit depends on the homophily of the graph—whether connected nodes share similar labels or behavior—and that heterophilic structure can hinder node-classification GNNs [18].

In the time-series anomaly detection literature, a recurring issue is the absence of a consistent distinction between faults and attacks: many studies treat these terms interchangeably or collapse them into a single "anomaly" class. A simulation study on a gasoline-factory segment using Modelica data [19] does separate certain faults that do not require expert analysis, motivating differentiated treatment of attacks. To control the false-positive rate and improve detection quality, several strategies have been proposed, including dynamic thresholds learned from denoising autoencoders [20] and stacked sparse autoencoders applied to fault diagnosis of rotating machinery [21]. These methods, however, focus on detection-side improvements; they do not characterize why a flagged time point is ambiguous, nor whether the underlying cause is a transient fault or a deliberate attack.

Graph-based anomaly detection has emerged as a way to capture inter-sensor structure that scalar or sequence models miss. In ICS sensor-fault detection, E-SFD [22] models sensor relationships to attribute faults to specific sensors, and Anomal-E [23] demonstrates self-supervised graph-based intrusion detection at the network level. These approaches

confirm that graph structure improves detection and per sensor attribution, but they remain detection-oriented and do not address downstream categorization of anomalies or correlation-aware interpretation across sensors.

As models grow more complex and behave as black boxes [8], XAI research has expanded to interpret their judgments [24]. Feature-attribution methods such as LIME [7] and SHAP [13,14] compute per-feature contributions and have been applied to industrial settings via custom local explainers such as CLE [25]. A complementary line of work generates representative patterns and measures pattern similarity to interpret the presence and degree of anomalies in each feature [26,27]. These approaches share two structural limitations relevant to ICS: they explain each sensor in isolation, and they do not natively expose temporal correlations between sensors. As a result, when an attack on one sensor propagates through correlated sensors, feature attribution surfaces all affected sensors at once but cannot reveal the dependency chain that drives expert response.

Three gaps therefore remain across the literature reviewed above. First, anomalies are detected but rarely categorized into actionable types such as fault versus attack: neither autoencoder-based detectors [20,27] nor graph-based ICS detectors such as E-SFD [22] and Anomal-E [23] draw this distinction, and even works that simulate cyber-attacks on industrial processes [19] report faults and attacks under a unified anomaly label. Second, false alarms near the detection threshold are mitigated by dynamic thresholding [20], but their systematic identification and interpretation as false positives or false negatives are not addressed. Third, XAI for time-series anomaly detection, whether via LIME [7] and SHAP [13], industrial extensions such as CLE [25], or pattern-similarity methods [26], remains feature-isolation based and leaves sensor-to-sensor correlations unexplained. The framework proposed in this paper targets the three gaps simultaneously: it defines anomaly categories from correlation characteristics, identifies false alarms via ambiguity analysis at the detection threshold, and exposes sensor correlations through graph-based interpretation.

4. Proposed Approach

Building a dependable AI-based anomaly detector requires both interpretable decisions and a principled vocabulary for what counts as "anomalous." Current XAI techniques explain individual feature contributions but rarely capture causal relationships in time-series data, and the lack of a consistent definition of anomaly types forces analysts to triage every detection from scratch.

This section proposes an approach that addresses both gaps. We categorize detected events into four types—faults, attacks, false positives, and false negatives—based on correlation characteristics, and we introduce graph-based interpretation methods that expose sensor correlations at decision points. Together, the categorization and interpretation layers identify the specific time points and sensor sets that warrant expert intervention.

4.1. Overview

As illustrated in Figure 1, the framework consists of an upstream detector and the proposed post-detection layer. The detector is the Graph Deviation Network (GDN) [28]: at each time point it forms a top-K correlation graph over the sensor embeddings, forecasts each sensor through graph attention, and flags anomalies from the per sensor deviation between predicted and observed values. This stage relies on standard techniques and is not a contribution of this work. The proposed post-detection layer then consumes the detector's outputs—the anomaly flags, the per sensor error scores, and the learned correlation graph—to (i) count the anomalous sensors among the top-K before and after smoothing, (ii) categorize each flagged time point as a fault, attack, false positive, or

false negative using the anomalous-sensor ratio before and after smoothing (which operationalizes correlation maintenance over the top-K correlated sensors), and (iii) produce a correlation-aware, category-specific interpretation. The novelty of this work resides entirely in this post-detection layer, which is agnostic to the specific upstream detector.

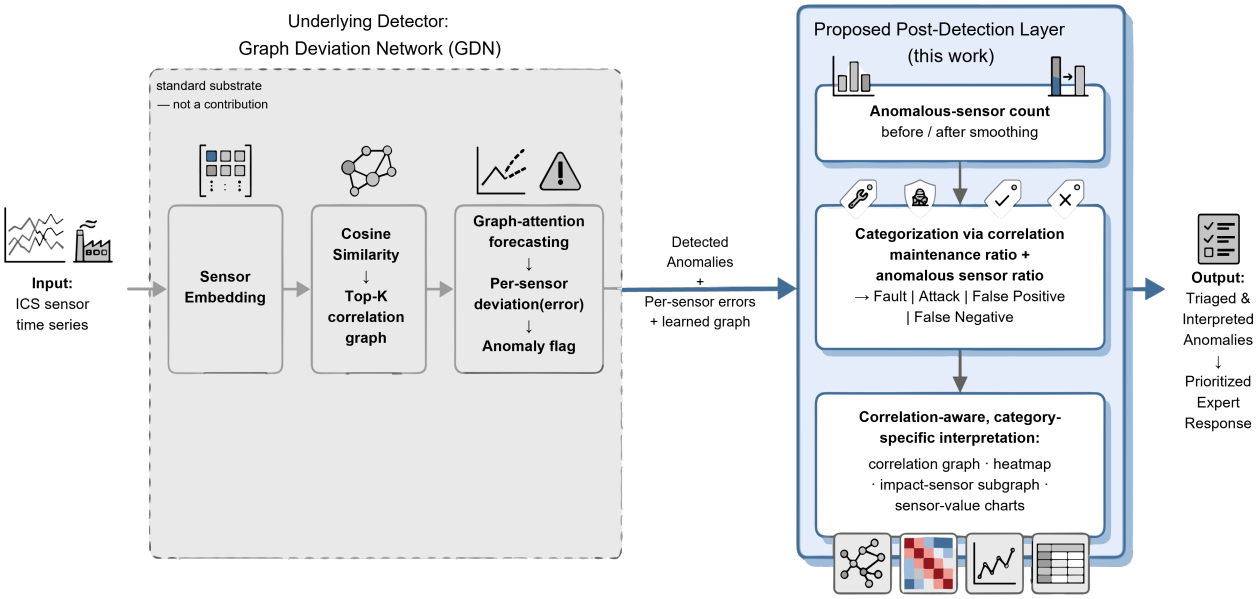


Figure 1. Overall framework. An upstream Graph Deviation Network (GDN) detector (**left**, dashed grey box) produces anomaly flags, per sensor errors, and the learned top-K correlation graph; the proposed post-detection layer (**right**) consumes these to categorize each anomaly as fault, attack, false positive, or false negative and to produce correlation-aware interpretation. The detector is a standard substrate and is not a contribution of this work.

4.2. Anomaly Types Definition

The term "anomaly" often encompasses both faults and attacks, yet in scenarios where attacks can result in severe consequences requiring immediate response, a comprehensive anomaly detection approach that includes both may prove inefficient. Moreover, attacks can precipitate unexpected faults. To differentiate between the two, this section provides examples and uses distinct characteristics to define types.

In fault scenarios, the focus is on undesirable events, failures, or quality deterioration. For example, in a water treatment setting, abrupt environmental changes such as rapid temperature fluctuations can cause anomalies in temperature sensors, possibly affecting pressure or water quality sensors. Automated systems typically adjust parameters to rectify such anomalies. If unsuccessful, expert intervention may be necessary. Characteristics of faults include the maintenance of correlations and a higher count of anomalous sensors.

Conversely, attacks involve intentional disruptions by external elements, hindering normal operation. In the water treatment example, an external intrusion might deliberately alter a specific temperature sensor's reading while keeping other sensors normal. This results in isolated anomalies, challenging the automated system. If persistent, this can lead to critical faults, necessitating expert intervention. Characteristics of attacks include the absence of correlation maintenance and fewer anomalous sensors.

These characteristics also aid in identifying false alarms. False positives, where normal conditions are mistakenly flagged as anomalies, are discerned by comparing them with genuine anomalies; false negatives, where anomalies are incorrectly deemed normal, are identified by contrasting them with actual normal instances. False positives tend to exhibit a low count of anomalous sensors after smoothing, low anomaly scores, and low correlation values, in contrast to the initially elevated pre-smoothing count that prompted the detection;

false negatives, by contrast, exhibit a high count of anomalous sensors both pre- and post-smoothing, high anomaly scores, and high correlation values.

To make these criteria precise, we formalize the anomalous-sensor ratio and the anomaly score over the top-K key sensors at each time point t. Writing $Z_i^{(t)}$ and $\tilde{Z}_i^{(t)}$ for the normalized per sensor error score before and after smoothing and τ_i for the per sensor anomaly threshold (all defined in Section 4.3), each key sensor is flagged before and after smoothing by

$$a_i^{(t)} = \mathbf{1}\Big[Z_i^{(t)} > \tau_i\Big], \qquad \tilde{a}_i^{(t)} = \mathbf{1}\Big[\tilde{Z}_i^{(t)} > \tau_i\Big], \tag{1}$$

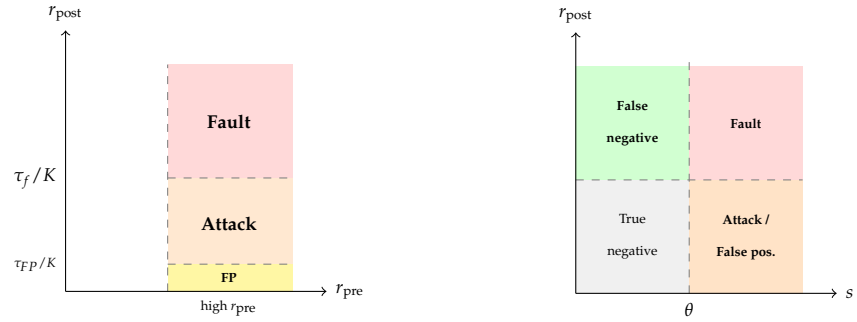
where $\mathbf{1}[\cdot]$ is the indicator function. The anomalous sensor ratio is the fraction of key sensors flagged as anomalous, evaluated before and after smoothing:

$$r_{\text{pre}}^{(t)} = \frac{1}{K}\sum_{i=1}^{K} a_i^{(t)}, \qquad r_{\text{post}}^{(t)} = \frac{1}{K}\sum_{i=1}^{K} \tilde{a}_i^{(t)} \in [0,1]. \tag{2}$$

Let $A^{(t)} = \{\, i : \tilde{a}_i^{(t)} = 1 \,\}$ be the key sensors that remain anomalous after smoothing. Because the top-K sensors are, by construction, the most mutually correlated peers selected from the learned embeddings, the size of $A^{(t)}$—equivalently $r_{\text{post}}^{(t)}$—serves as a direct, computable proxy for correlation maintenance: when a disturbance propagates through coupled sensors (a fault), several of these correlated sensors remain jointly anomalous and $r_{\text{post}}^{(t)}$ stays high, whereas an isolated manipulation (an attack) leaves only one or a few, collapsing $r_{\text{post}}^{(t)}$. The underlying pairwise correlations $c_{ij}^{(t)}$ between sensors i and j are not thresholded as a separate decision variable; instead they are displayed directly as correlation heatmaps in the case studies (Section 5) to corroborate each categorization. Finally, $s^{(t)}$ denotes the aggregate anomaly score on which detection is based, so that the detector predicts time point t as anomalous when $s^{(t)} > \theta$.

Given these quantities, the four categories are separated by the decision rules in Table 1, which operate on the pre- and post-smoothing anomalous-sensor ratios. Among the points the detector flags as anomalous, a fault retains a high post-smoothing ratio while an attack collapses to a few anomalous sensors after smoothing; a false positive is the limiting case whose post-smoothing ratio drops to its lowest, and is therefore identified as the subset of the attack pool with the smallest residual count. Among the points predicted to be normal, a false negative is one that nonetheless retains a high anomalous-sensor ratio both before and after smoothing. The thresholds τ_f and τ_a (post-smoothing fault and attack counts), τ_p (pre-smoothing count), and τ_{FP}, τ_{FN} (post-smoothing false-alarm counts), together with the detection threshold θ, are calibrated empirically; the specific values used in our experiments are reported in Section 5. The separation of faults from attacks rests on the canonical pattern in which faults preserve inter-sensor correlation while attacks break it; boundary cases that can violate this assumption—stealthy multi-point attacks and localized single-sensor faults—are discussed in Section 6.

Figure 2 visualizes these rules as decision regions. Part (a) plots the anomalous points in the plane of the pre- and post-smoothing ratios $r_{\text{pre}}^{(t)}$ and $r_{\text{post}}^{(t)}$: the detector's anomalies share a high pre-smoothing ratio, and the post-smoothing ratio then separates faults (which stay high) from attacks and false positives (whose count collapses). Part (b) uses the anomaly score $s^{(t)}$ and $r_{\text{post}}^{(t)}$, with the detection threshold θ separating model-normal from model-anomalous points and thereby exposing false negatives. The operational procedure for measuring the pre- and post-smoothing counts—in particular, the dynamic threshold smoothing—is detailed in Section 4.3.



(**a**) (**b**)

Figure 2. Decision regions for the four anomaly categories (cf. Table 1). (**a**) Detector-flagged (anomalous) points share a high pre-smoothing ratio r_{pre}; the post-smoothing ratio r_{post} then separates faults ($r_{\text{post}} \geq \tau_f/K$, the correlated sensors stay jointly anomalous) from attacks ($r_{\text{post}} \leq \tau_a/K$, the count collapses), with the false positives being the subset whose residual count is smallest ($r_{\text{post}} \leq \tau_{FP}/K$). (**b**) In the plane of anomaly score s and r_{post}, the detection threshold θ separates model-normal points (false negatives at high r_{post} that also satisfy $r_{\text{pre}} \geq \tau_p/K$; true negatives otherwise) from model-anomalous points (faults at high r_{post}; attacks and false positives at low r_{post}). The pairwise correlations underlying the fault/attack distinction are shown directly via the heatmaps of Section 5. Empirical threshold values are given in Section 5.

Table 1. Decision rules separating the four anomaly categories from the detector's prediction and the pre-/post-smoothing anomalous-sensor ratios of Equation (2). Thresholds are expressed as sensor counts over the K key sensors; the empirical values are given in Section 5. The false-positive criterion ($\tau_{FP} \leq \tau_a$) isolates, within the low-post-count attack pool, the points whose anomalous signal collapses furthest after smoothing.

Model Prediction	Condition	Category
Anomalous ($s^{(t)} > \theta$)	$r^{(t)}_{\text{post}} \geq \tau_f/K$	Fault
Anomalous ($s^{(t)} > \theta$)	$r^{(t)}_{\text{post}} \leq \tau_a/K$	Attack
Anomalous ($s^{(t)} > \theta$)	$r^{(t)}_{\text{post}} \leq \tau_{FP}/K$ (subset of Attack)	False positive
Normal ($s^{(t)} \leq \theta$)	$r^{(t)}_{\text{pre}} \geq \tau_p/K$ and $r^{(t)}_{\text{post}} \geq \tau_{FN}/K$	False negative
Normal ($s^{(t)} \leq \theta$)	otherwise	True negative

4.3. Detection and Interpretation Pipeline

The graph-extraction, per sensor scoring, and smoothing stages described in this subsection follow established practice in graph-based time-series anomaly detection and are not, in themselves, claimed as novel. Concretely, we adopt the Graph Deviation Network (GDN) [28], a graph-attention reconstruction model, as the underlying detector. They are presented for completeness because they produce the substrate—per-time-point correlation graphs, per sensor anomaly scores, and pre- and post-smoothing anomalous-sensor counts—on which the categorization and interpretation layer (the actual contribution of this work) operates. The layer is agnostic to the specific detector and could be paired with an alternative upstream model.

4.3.1. Correlation and Graph Extraction

At each time point, the framework identifies the top-K sensors with the strongest mutual correlations and builds a subgraph from them. The input data first pass through an embedding layer that produces an initial vector for each sensor. Cosine similarities between these vectors are then computed via matrix multiplication, and the K sensors with the strongest mutual similarities are retained as the nodes of the subgraph at that time point,

with edges between them weighted by their pairwise cosine similarities. Repeating this construction across all time points yields the full graph. The resulting subgraphs and the overall graph structure expose the system architecture and the inter-sensor relationships at each examined time point.

4.3.2. Per-Sensor Anomaly Scoring

For each key sensor i and time point t, the framework compares the actual sensor value $S_i^{(t)}$ with the model's prediction $\widehat{S}_i^{(t)}$. The per sensor error score, defined in Equation (3), is the absolute difference between the two:

$$\text{Err score}_i^{(t)} = \left| S_i^{(t)} - \widehat{S}_i^{(t)} \right| \tag{3}$$

Because sensors have different value distributions and error scales, the raw error scores must be normalized so that the detection is not biased toward any single sensor. Equation (4) applies this normalization, where $\overline{p}_i$ and $\overline{o}_i$ denote the median and interquartile range of $\text{Err score}_i^{(t)}$, respectively.

$$Z_i^{(t)} = \frac{\text{Err score}_i^{(t)} - \overline{p}_i}{\overline{o}_i} \tag{4}$$

4.3.3. Error Score Smoothing

Anomaly detection over the top-K key sensors can produce excessive false positives. To mitigate this, we apply dynamic threshold smoothing that adapts to the test data. Per-sensor thresholds are placed near the top-0.05% quantile, following common practice in time-series anomaly detection [1], and are obtained as the mean of heuristic thresholds derived from the training and test data.

Smoothing is applied to the per sensor error scores so that transient spikes that do not reflect a sustained anomaly are suppressed before thresholding. The effect of smoothing is then read off the count of key sensors flagged as anomalous at each time point: comparing the pre- and post-smoothing counts together characterizes the four anomaly categories defined in Section 4. A high count both pre- and post-smoothing indicates a fault. A low count post-smoothing indicates an attack. An initially high count that drops after smoothing flags a candidate false positive. Conversely, a normal-side prediction with persistently high counts both pre- and post-smoothing flags a candidate false negative.

4.3.4. Interpretation Method

The interpretation is assembled from the visual artifacts produced by the post-detection layer in Figure 1. The correlation graph built from the top-K sensors makes inter-sensor relationships explicit, while a correlation heatmap encodes the numerical strength of those relationships—red indicates high correlation, blue indicates low correlation. Together, the heatmap and the impact-sensor subgraph allow an analyst to read off the sensors most affected at the time point under analysis and to compare their actual values with those observed during normal and anomalous periods.

The resulting picture pinpoints where intervention is needed. For faults, the responsible sensor and its correlated neighbors can be identified together, exposing the causal chain. For attacks, additional sensors that may be affected downstream become visible through the correlation edges, narrowing the set of sensors that warrant further inspection.

5. Experiment

This section presents the experimental results of applying the proposed approach. The focus is on identifying the four anomaly categories—faults, attacks, false positives,

and false negatives—and interpreting representative cases through visualizations and numerical data.

The experiments were conducted on an Intel Xeon Gold 6226 2.7 GHz server with 128 GB of RAM and an NVIDIA Tesla T4 16 GB GPU, using Python 3.11.5 within Anaconda 3 and Jupyter Notebook 7.0.6.

5.1. Dataset

The SWaT dataset, collected by iTrust (Centre for Research in Cyber Security) and SUTD (Singapore University of Technology and Design) from a water treatment test platform, was employed for the experiments [29]. The composition of the dataset is detailed in Table 2. The training data encompass all normal operating data, amounting to 991,800 indices (about 275.5 h in total, Table 2). The test data consist of 36 abnormal scenarios, totaling 449,909 indices (about 125 h). Overall, the dataset features a total of 51 attributes [30].

Table 2. Dataset composition.

Dataset	Files	Attack Counts	Interval (h)	Size (MB)
Train	train0.csv	–	138	142
	train1.csv	–	137.5	145
	total	–	275.5	287
Test	test0.csv	36	125	130
	total	36	125	130

5.2. AI Model

We emphasize at the outset that the contribution of this work lies in the categorization and interpretation layer that runs on top of the detector, rather than in raw detection accuracy. Accordingly, the detection results reported in this subsection serve as the input substrate for the identification and interpretation analyses presented in the subsequent subsections, and are not intended as a head-to-head benchmark against detection-focused baselines. In short, this paper concerns how a detector's outputs are categorized, triaged, and interpreted—not whether the detector itself advances raw detection accuracy.

Models with three different top-*K* parameters were developed using a graph-based approach and subsequently evaluated for performance. The training and validation ratios for all models were established at 8:2. Performance results for top-*K* parameters of 5, 10, and 15 are tabulated in Table 3. The model with the top-*K* parameter set to 10 demonstrated optimal performance and was used as the underlying detector for the identification and interpretation of anomalies by type. Throughout this paper we use the standard confusion-matrix terminology: a true positive (TP) is an actual anomaly correctly flagged as anomalous; a false positive (FP) is a normal time point incorrectly flagged as anomalous; a true negative (TN) is a normal time point correctly identified as normal; and a false negative (FN) is an actual anomaly that the model incorrectly classifies as normal. Table 4 summarizes these counts on the test set. Out of a total of 449,909 test indices, 54,621 are actually anomalous and 395,288 are actually normal. The AI model flagged 40,491 indices as anomalous—comprising 37,426 true positives and 3065 false positives—and 409,418 indices as normal, of which 17,195 are false negatives (actual anomalies missed by the model).

Table 3. AI model performance metrics across top-*K* (SWaT).

Methods	TopK = 5	TopK = 10	TopK = 15
F1 Score	0.738	0.787	0.74
Precision	0.912	0.925	0.99
Recall	0.62	0.685	0.59

To confirm that these results are stable across random initializations, we also evaluated the Top-$K = 10$ model over five random seeds, obtaining a precision of 0.925 ± 0.008, a recall of 0.685 ± 0.010, and an F1-score of 0.787 ± 0.009 (mean $\pm$ 95% confidence interval). The narrow intervals indicate robustness to stochastic training variance; these figures characterize the reliability of the substrate on which the interpretation layer operates, not a competitive detection benchmark.

Table 4. AI model anomaly detection summary.

Total	Normal	Attack	Prediction	
			Normal	Attack
449,909	395,288	54,621	409,418	40,491
True positives		**False negatives**		**False positives**
37,426		17,195		3065

After anomaly detection, each time point is categorized into one of the four anomaly types defined in Section 4:

1. For every time point, we examine the per sensor anomaly status of the top-K key sensors, both before and after applying smoothing. In this experiment $K = 10$, so the analysis at each time point focuses on its top 10 sensors.
2. Categorization then uses the count of anomalous sensors before and after smoothing, which serves as a proxy for the preservation of correlations among the top-K sensors. Time points predicted as normal by the model yield candidate false negatives, while time points predicted as anomalous yield candidates for false positives, faults, and attacks.

Concretely, the symbolic thresholds of the decision rules in Table 1 take the following values in our experiments ($K = 10$). Among the detector-flagged points, the fault/attack separation is placed on the post-smoothing count at $\tau_f = 3$ (fault) and $\tau_a = 2$ (attack), which partition the anomalous pool; the false positives are the subset whose post-smoothing count collapses to $\tau_{FP} = 1$. False negatives, drawn from the points predicted normal, require a saturated pre-smoothing count $\tau_p = 10$ (i.e., $r_{\text{pre}} = 1.0$) together with a post-smoothing count of at least $\tau_{FN} = 5$. The sensitivity of the identification results to these threshold choices is examined later in this section.

5.3. Robustness and Sensitivity Analysis

Because the categorization thresholds in Section 4 are calibrated empirically, we examine how sensitive the framework is to its main hyperparameters. Table 5 varies the temporal window size and Table 6 sweeps the detection-threshold percentile on SWaT; the top-K comparison was given in Table 3.

Table 5. Window-size sensitivity (Top-$K = 10$, stride 1) on SWaT.

Window	Prec.	Rec.	F1
5	0.952	0.647	0.770
10	0.925	0.685	0.787
20	0.913	0.658	0.765

The framework is stable in the neighborhood of its operating point—around the chosen window size and top-K, the F1-score varies by only a few points. The detection threshold, however, is calibrated empirically (the optimum lies near the 91st percentile here) and

would require recalibration in a different ICS environment rather than being transferred unchanged, which we note among the limitations.

Table 6. Threshold-percentile sensitivity on SWaT. The optimal operating point lies near the 91st percentile.

Percentile	Prec.	Rec.	F1
50th	0.214	0.883	0.345
60th	0.255	0.840	0.391
70th	0.323	0.799	0.460
80th	0.434	0.715	0.540
90th	0.845	0.696	0.763
95th	0.973	0.401	0.568
97th	0.968	0.239	0.384
99th	0.922	0.076	0.140
Optimal	0.925	0.685	0.787

Because the contribution of this work is the post-detection layer rather than the detector, the most decision-relevant ablation is the effect of the smoothing and threshold steps on false-alarm identification, reported directly in Section 5: the pre- versus post-smoothing comparison in Figure 3 and the threshold sweeps in Tables 7 and 8 isolate how much each step contributes to identifying false negatives and false positives. The categorization step itself is post hoc and does not alter the detection metrics; its contribution is measured by the false-negative and false-positive identification precision reported there. Detector-internal choices such as the graph construction are part of the substrate and are reflected in the top-K sensitivity above.

Normal Time Index Before Smoothing	
Anomaly Sensor counts	Time Index Counts
3	459
4	5092
5	28,432
6	90,464
7	114,893
8	104,212
9	51,951
10	13,915

Normal Time Index After Smoothing	
Anomaly Sensor Counts	Time Index Counts
0	**332,751**
1	**68,555**
2	6359
3	1148
4	302
5	164
6	137
7	2

(a) Normal Time Points

Anomaly Time Index Before Smoothing	
Anomaly Sensor counts	Time Index Counts
4	1
5	19
6	38
7	595
8	1685
9	7038
10	31,115

Anomaly Time Index After Smoothing	
Anomaly Sensor Counts	Time Index Counts
1	328
2	4250
3	**14,109**
4	**13,867**
5	**6025**
6	**515**
7	**391**
8	**291**
9	**337**
10	**378**

(b) Anomaly Time Points

Figure 3. Impact of smoothing on time points categorized as normal and anomalous by the AI model.

Table 7. False-negative identification across pre- and post-smoothing count thresholds. The bolded row corresponds to the criterion adopted in this study (10 pre-smoothing, 5 post-smoothing), yielding 248 flagged points with 83% precision.

Pre-Smoothing	Post-Smoothing	Total Flagged	Correct	Incorrect	Precision
9	4	528	227	301	43%
9	5	300	212	88	71%
10	4	300	212	88	71%
10	**5**	**248**	**207**	**41**	**83%**

Table 8. False-positive identification across pre- and post-smoothing count thresholds. The bolded row corresponds to the criterion adopted in this study (10 pre-smoothing, 1 post-smoothing), yielding 328 flagged points with 73% precision.

Pre-Smoothing	Post-Smoothing	Total Flagged	Correct	Incorrect	Precision
9	1	211	132	79	63%
9	2	3875	864	3011	22%
10	**1**	**328**	**240**	**88**	**73%**
10	2	4578	1419	3159	31%

5.4. Identification

Figure 3 shows the effect of dynamic-threshold smoothing on time points the AI model predicted as normal and as anomalous. Before smoothing, both groups exhibit elevated counts of anomalous sensors among the top 10, with the normal group only slightly lower; this weak separation contributes to false positives. After smoothing, the two groups become clearly distinguishable, as highlighted by the blue boxes: normal time points typically show 0 or 1 anomalous sensors, whereas anomalous time points show 3 or more.

The histograms in Figure 3 make the effect of smoothing quantitatively visible. For normal-classified time points (Figure 3a), the dominant pre-smoothing counts of 7 and 8—at 114,893 and 104,212 time points, respectively—collapse after smoothing into counts of 0 (332,751 points) or 1 (68,555 points), reflecting the suppression of transient noise that would otherwise inflate the false-positive rate. The anomaly-classified group shows the opposite pattern (Figure 3b): pre-smoothing counts are concentrated at the high end (31,115 points with 10 anomalous sensors), but smoothing redistributes the mass across a range from 1 to 10, with the largest concentrations at counts of 3 and 4 (14,109 and 13,867 points). The two distributions cease to overlap meaningfully at counts of 3 or more, which justifies using the post-smoothing count as the principal discriminator between the four anomaly categories. The long tails—where anomaly-classified time points retain low post-smoothing counts (1–2) or normal-classified time points retain high counts ($\geq$4)—are exactly the regions in which the four categories defined in Section 4 emerge as identifiable cases, examined individually in the three subsubsections below.

5.4.1. Case-1: False-Negative Identification

False negative candidates are extracted from Figure 3a, which corresponds to time points predicted as normal by the model. Before smoothing, 65,866 of these points already had 9 or 10 anomalous sensors among the top 10. After smoothing, points retaining 4 or more anomalous sensors are prioritized as candidate false negatives. Table 7 reports correct and incorrect identifications across several count thresholds. Using a criterion of 10 pre-smoothing and 5 post-smoothing, 248 candidate points were flagged: 207 were confirmed as actual anomalies (correct identifications) and 41 as normal (incorrect identifications), giving an 83% precision in false-negative identification.

The four row entries in Table 7 make the trade-off between the two thresholds explicit. The (9, 4) criterion is the most permissive, flagging 528 candidates at only 43% precision because it admits many near-threshold cases whose post-smoothing count of 4 is too weak to discriminate true anomalies from noise. Tightening either threshold individually yields the (9, 5) and (10, 4) variants, both of which collapse to 300 flagged candidates at 71% precision—an indication that the two thresholds carry comparable discriminative power in isolation. Requiring both simultaneously, in the chosen (10, 5) criterion, further trims the flagged set to 248 candidates and lifts precision to 83%, showing that the high-confidence false-negative signature is the simultaneous persistence of both pre- and post-smoothing evidence.

5.4.2. Case-2: False-Positive Identification

False-positive candidates are extracted from Figure 3b at time points predicted as anomalous by the model. Before smoothing, 38,153 such points have 9 or 10 anomalous sensors. After smoothing, points whose count drops to 2 or fewer are prioritized as candidate false positives. Table 8 reports correct and incorrect identifications across count thresholds. Using a criterion of 10 pre-smoothing and 1 post-smoothing, 328 candidate points were flagged: 240 were confirmed as normal (correct identifications) and 88 as actual anomalies (incorrect identifications), giving roughly 73% precision in false-positive identification.

The four row entries in Table 8 show that the post-smoothing count is the dominant axis for this case. With a post-smoothing count of 2, the candidate set inflates to 3875 or 4578 points at only 22–31% precision, because the criterion now also captures attack-classified time points that retain a small anomalous-sensor signature. With a post-smoothing count of 1, the candidate set is much tighter and precision rises sharply. The (10, 1) criterion adopted in this study is therefore the cleanest operating point for FP-specific identification: requiring the post-smoothing count to collapse to a single residual sensor isolates the time points whose anomaly signal does not persist after smoothing—the signature of a transient detector spike rather than a sustained anomaly.

5.4.3. Case-3: Attack/Fault Identification

Among the points predicted as anomalous, we further separate attacks from faults using the criteria defined in Section 4: faults retain high correlations and a high count of anomalous sensors both before and after smoothing, whereas attacks exhibit low correlations after smoothing and only 1–2 anomalous sensors. In Figure 3b, the blue-boxed regions correspond to faults, and the regions with 1 or 2 anomalous sensors correspond to attacks. The dataset is thus split into 35,913 fault instances and 4578 attack instances, providing a prioritized basis for downstream interpretation.

The split is highly asymmetric: faults account for 88.7% of the predicted anomalies and attacks for the remaining 11.3%. Concentrating expert attention on the smaller attack pool is consistent with the availability-first posture of ICS operation, since the larger fault pool is dominated by transient deviations that automated controllers typically absorb, whereas the attack pool is where critical-impact cases concentrate. The attack pool also overlaps deliberately with the false-positive candidates of Case-2, because both populations share a low post-smoothing count; the 328 false-positive candidates flagged earlier are therefore drawn from the 4578 attack-classified points rather than from the fault pool. The two layers thus serve complementary roles—Case-2 separates the genuine normal events that slipped past the detector from the candidate attacks, while Case-3 separates the remaining anomaly pool into actionable attack candidates and routine faults.

5.5. Interpretation

For each identified case, we interpret it by combining five pieces of information at the time point of interest:

1. Err score and anomaly status of the top 10 sensors, indicating how far each sensor deviates from its prediction and whether it is flagged as anomalous;
2. Subgraph of key sensors, visualizing the inter-sensor structure at that time point;
3. Correlation heatmap of the top 10 sensors, encoding the numerical strength of inter-sensor correlations;
4. Impact-sensor analysis, extending from the anomalous sensors to neighbors that may be influenced through correlation edges;
5. Sensor-value visualization, comparing the actual readings of the key sensors at the time point against their normal-period values.

5.5.1. Case-1: Interpretation of False-Negative Cases

At time point 436,577, which the AI model predicted to be normal but which is actually anomalous, we examined the top 10 key sensors: MV101, FIT301, FIT401, FIT502, FIT501, PIT503, PIT501, FIT503, FIT504, and MV304. Their subgraph is shown in Figure 4.

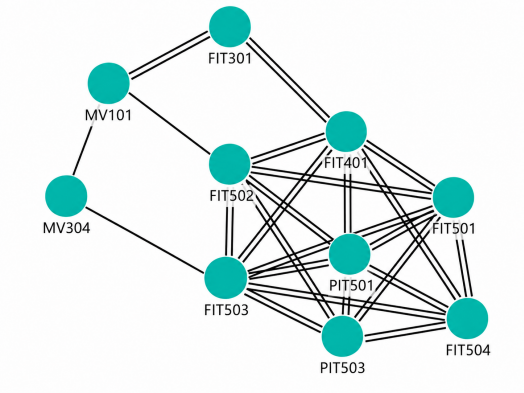


Figure 4. Impact sensor subgraph (false negatives).

The subgraph centers on PIT501, identifying it as the most influential sensor at this time point. The correlation heatmap in Figure 5 (red: high correlation, blue: low correlation) confirms that most of the top 10 sensors are strongly correlated. Before smoothing, all ten sensors were flagged as anomalous; after smoothing, five remained: PIT503, PIT501, FIT503, FIT504, and MV304.

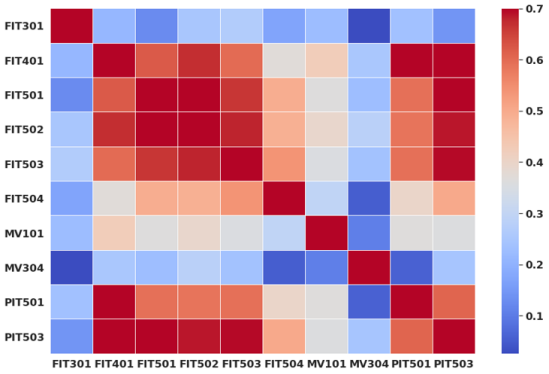


Figure 5. Correlation coefficient heatmap (false negatives).

Figure 6 visualizes the impact sensors centered on these five anomalous nodes. Red nodes are the flagged sensors, blue nodes are additional sensors of interest reachable through correlation edges, and green nodes are sensors that connect into the flagged ones. The same color convention is used for all impact-sensor graphs; green nodes therefore appear only when sensors connect into the flagged ones, so in cases where no such sensor

exists the graph contains only red and blue nodes. Beyond the five anomalous sensors, three further sensors (FIT501, FIT401, FIT502) shown as blue nodes warrant follow-up analysis.

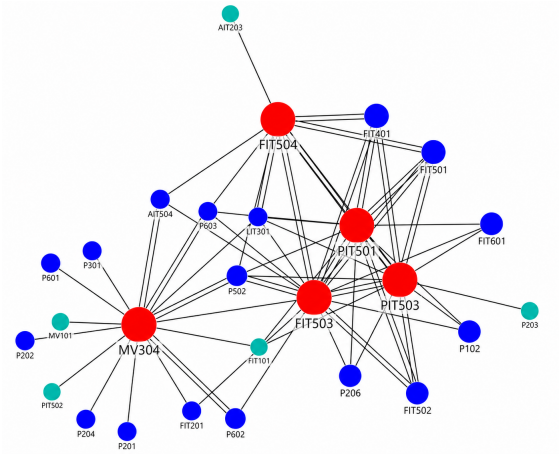


Figure 6. Anomalous sensor impact graph (false negatives).

According to the dataset annotations, the anomaly at this time point was induced by manipulation of the P501 and FIT502 sensors, producing a fault. Figure 7 compares the actual values of the five identified anomalous sensors with their normal-period values: PIT501 and PIT503 in particular deviate sharply from normal and then recover, while the remaining three sensors show preceding changes consistent with the propagation of the induced fault.

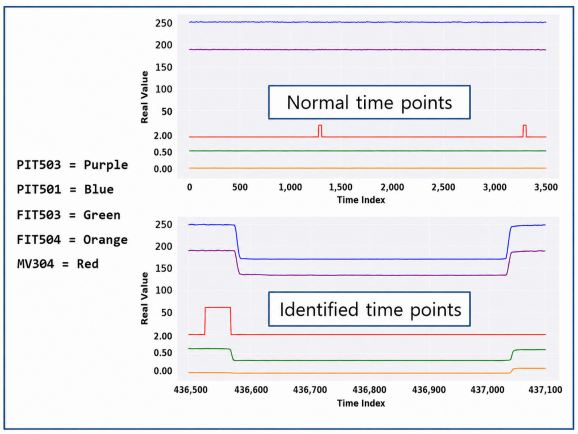


Figure 7. Anomalous sensor value comparison (false negatives).

5.5.2. Case-2: Interpretation of False Positives Case

At time point 264,349, which the AI model initially flagged as anomalous but which has been re-classified as normal under the criteria of Section 4, the top key sensors are FIT201, MV201, AIT504, PIT502, MV101, AIT203, P502, MV304, and P201. Their subgraph is shown in Figure 8.

The subgraph centers on the P201 sensor, identifying it as the most influential at this time point. The correlation heatmap in Figure 9 shows that, in contrast to the fault case, most sensors are marked in blue, confirming low correlations. Before smoothing, all of these key sensors were flagged as anomalous; after smoothing, only P201 remained flagged.

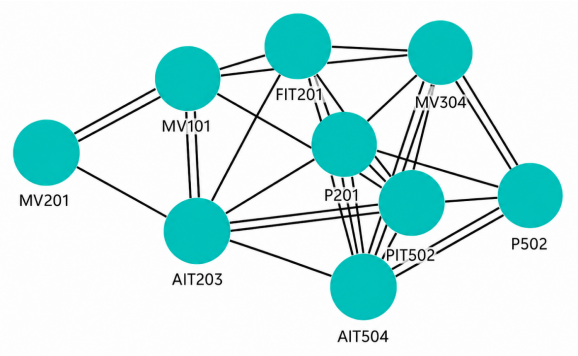


Figure 8. Impact sensor subgraph (false positives).

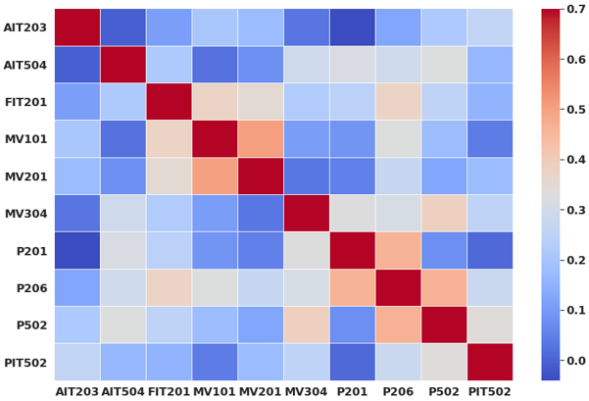


Figure 9. Correlation coefficient heatmap (false positives).

Figure 10 visualizes the impact sensors around this single anomalous node, following the same color scheme (red: flagged sensor; blue: additional sensors of interest; green: sensors connecting into the flagged one). Beyond P201, three additional sensors shown as blue nodes (P202, P404, AIT504) can be selected for follow-up. Figure 11 compares the actual values of P201 with its normal-state distribution and shows a pattern that closely matches normal behavior, confirming a false-positive detection.

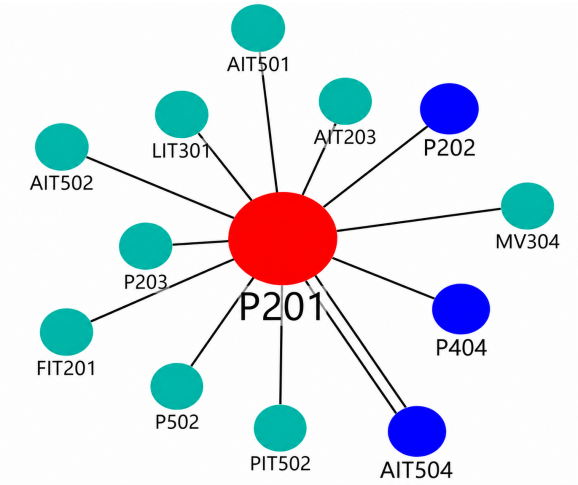


Figure 10. Anomalous sensor impact graph (false positives).

According to the dataset annotations, this time point is in fact normal, validating the false-positive identification.

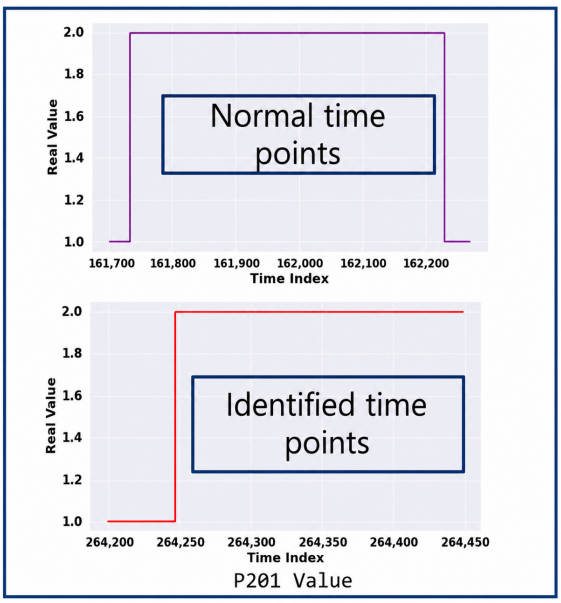


Figure 11. Anomalous sensor values comparison (false positives).

5.5.3. Case-3: Interpretation of Attack Case

At time point 7229, which the AI model flagged as anomalous and which our categorization classifies as an attack, the top 10 key sensors are PIT501, AIT504, FIT501, PIT502, P502, AIT202, FIT101, P203, AIT402, and FIT301. Their subgraph is shown in Figure 12.

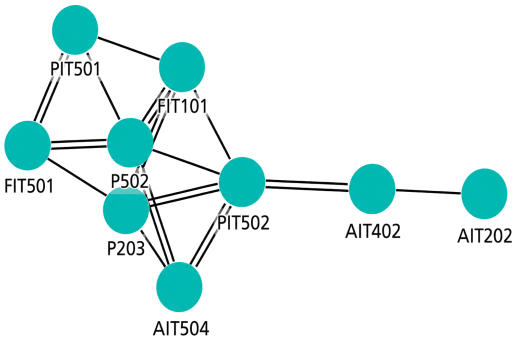


Figure 12. Impact sensor subgraph (attack).

The subgraph centers on P502 and PIT502, identifying them as the most influential sensors at this time point. The correlation heatmap in Figure 13 shows that, in contrast to the fault cases above, most sensors are marked in blue, indicating low correlations.

Before smoothing, nine sensors (all but PIT501) were flagged as anomalous; after smoothing, only AIT402 and FIT301 remained flagged. Figure 14 visualizes the impact sensors around these two anomalous nodes: the two flagged sensors (AIT402, FIT301) are shown as red nodes and the correlated sensors reachable from them as blue nodes. Two additional sensors (AIT202, PIT502), which also appear in the subgraph of Figure 12, are selected for follow-up.

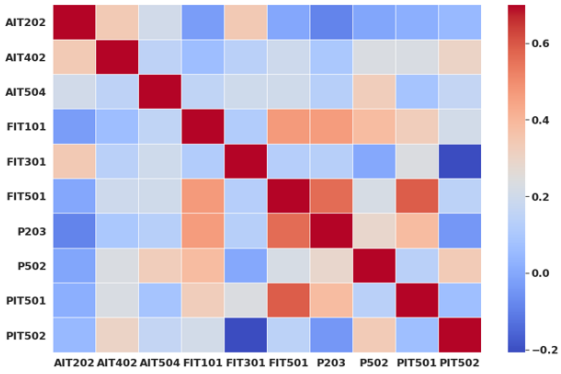


Figure 13. Correlation coefficient heatmap (attack).

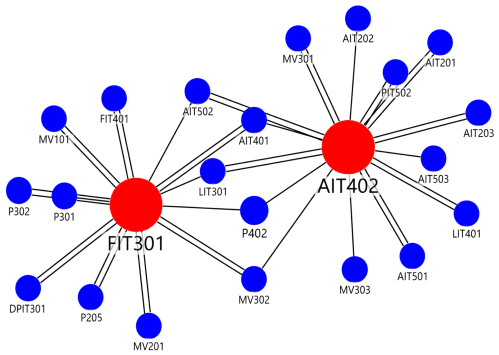


Figure 14. Anomalous sensor impact graph (attack).

According to the dataset annotations, this time point corresponds to an attack that manipulated the AIT202 reading; its impact propagated to the actuator P203 and produced unexpected downstream effects, with AIT402 and FIT301 becoming anomalous immediately and PIT502 and AIT504 affected about two hours later.

Figure 15 shows the actual values of AIT202 (the source of the attack) and of the two identified anomalous sensors AIT402 and FIT301, compared against their normal-state values. AIT202 (blue) drops sharply from its normal-state values (red) at the identified time point, confirming the manipulation. AIT402 and FIT301, while also deviating from normal, do so far less dramatically than AIT202—consistent with their role as downstream sensors affected by the attack rather than the manipulated source. The visualization thus makes the attack's propagation path explicit: AIT202 is the initiating sensor, and AIT402 and FIT301 are its downstream consequences.

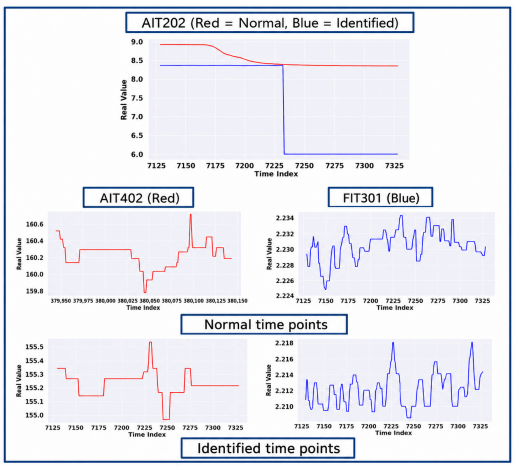


Figure 15. Anomalous sensor value comparison (attack).

5.5.4. Case-4: Interpretation of Fault Case

At time point 348,322, which the AI model flagged as anomalous and which our categorization classifies as a fault, the top 10 key sensors are P603, P101, FIT101, FIT201, P203, MV302, FIT301, P302, MV101, and P205. The subgraph in Figure 16 centers on MV101, MV302, P205, and P302. The correlation heatmap in Figure 17 shows that most sensors are marked in red, indicating high correlations among the top 10 sensors.

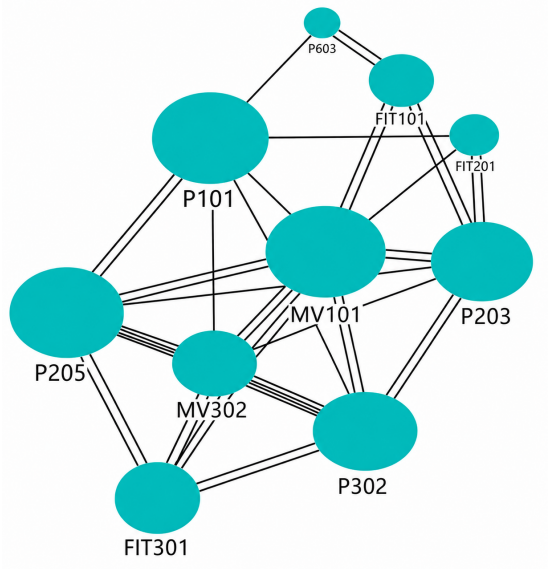


Figure 16. Impact sensor subgraph (fault).

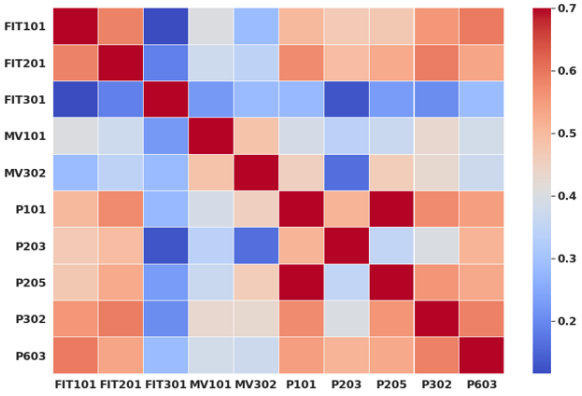


Figure 17. Correlation coefficient heatmap (fault).

Before smoothing, all ten sensors were flagged as anomalous; after smoothing, six remained flagged: P203, MV302, FIT301, P302, MV101, and P205. Figure 18 visualizes the impact sensors around these six anomalous nodes, with the flagged sensors shown as red nodes and the correlated sensors reachable from them as blue nodes. Three further sensors shown as blue nodes (P101, FIT101, FIT201), which also appear in the subgraph of Figure 16, can be selected for follow-up. According to the dataset annotations, the fault was induced by manipulating P302, with no anomalies beyond the expected propagation. Figure 19 compares the actual values of the six anomalous sensors against their normal-state values, showing significant decreases at the identified time point that confirm the fault. The subgraph, impact-sensor graph, and value comparison together provide an intuitive picture of the fault and pinpoint the sensors that warrant further analysis.

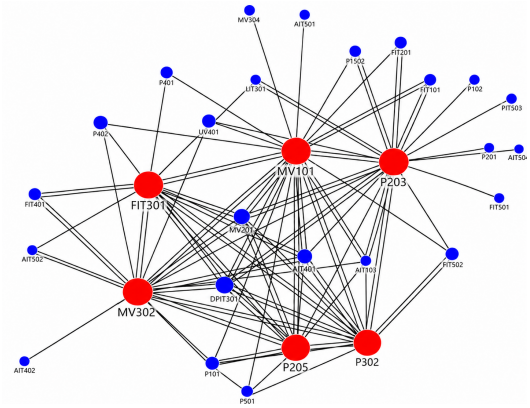


Figure 18. Anomalous sensor impact graph (fault).

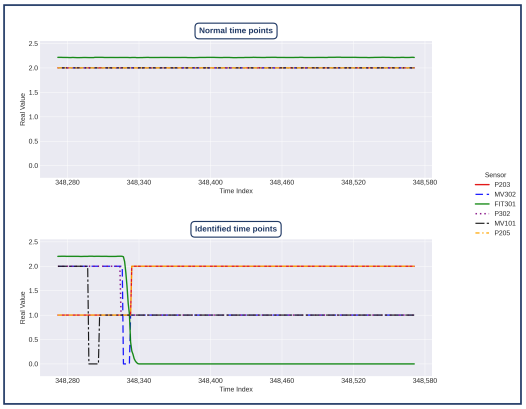


Figure 19. Anomalous sensor value comparison (fault).

6. Discussion

We applied the proposed framework to the SWaT dataset and used the pre- and post-smoothing anomalous-sensor ratios, together with the anomaly score, to separate faults, attacks, false positives, and false negatives from a single pool of detected events; the inter-sensor correlation structure that motivates the fault/attack distinction is then made explicit through the case-study heatmaps.

Three findings stand out. First, separating faults from attacks at detection time enables prioritized response: rather than triaging every flagged event identically, analysts can focus expert attention on the smaller subset of high-impact attack candidates. Second, the ambiguity analysis recovers candidate false negatives and false positives near the detection threshold with identification precision of 83% and 73%, respectively, (Tables 7 and 8); these candidates are concrete cases an expert can re-examine, and they can feed back into model retraining. Third, the graph-based interpretation makes sensor-to-sensor correlations explicit at decision points, going beyond per-feature attribution and aligning with the propagation patterns documented in the SWaT scenarios (see Case 3 in Section 5).

These outcomes were validated against the anomaly timestamps released with SWaT [29]: the categorization recovers the intended fault/attack distinctions, and the impact-sensor subgraphs correctly highlight the sensors influenced by manipulated inputs. By replacing fixed-threshold blanket triage with characteristic-based categorization, the framework provides a more credible basis for downstream expert review in availability-critical environments.

Graph homophily and the role of heterophily. A natural question is how the homophily or heterophily of the constructed graph affects the approach [18]. Two points are worth noting. First, the homophily principle of [18] concerns node classification, where

aggregating over dissimilar (heterophilic) neighbors can degrade label prediction; our framework does not classify node labels but uses the graph to reconstruct sensor values and, post hoc, to interpret inter-sensor structure, so this failure mode does not apply directly. In addition, the detector aggregates only over the top-K most correlated peers selected from the learned sensor embeddings, with attention weighting, rather than over arbitrary neighborhoods, which structurally limits the influence of weakly related sensors. Second, and more importantly, the homophily level is aligned with the signal the categorization exploits: because the top-K sensors are the most mutually correlated peers, the number that remain jointly anomalous after smoothing reflects how strongly coupled the disturbance is—a larger such set (homophilic) indicates a fault, whereas a collapse to one or a few (heterophilic or isolated) indicates an attack or a false alarm. A quantitative comparison with heterophily-specific graph neural networks would target the upstream detector rather than the post-detection layer that constitutes our contribution, and is left to future work.

Limitations

It is important to note that the contribution of this work lies in the categorization and interpretation layer that runs on top of an anomaly detector, rather than in raw detection accuracy. The following limitations should be acknowledged when assessing the framework.

Single-dataset validation and threshold calibration. The categorization criteria and the interpretation procedure are validated on the SWaT dataset, the de facto benchmark for ICS anomaly research; generalization to other testbeds such as WADI and HAI and to production ICS with different sensor counts, sampling rates, and process dynamics remains to be verified empirically. In addition, both the operating detection threshold and the category-level cutoffs that separate the Fault, Attack, and false-alarm regions are calibrated empirically on the training distribution (Table 6) rather than derived from a principled statistical procedure, so a different ICS environment would require recalibration rather than transfer of the values unchanged; automatic threshold adaptation is left to future work.

Boundary cases of the correlation-based assumption. The categorization rests on the assumption that faults preserve inter-sensor correlation while attacks break it, and two scenarios can violate it. A stealthy multi-point attack that manipulates several sensors in a coordinated manner can keep them mutually correlated and produce a high anomalous-sensor ratio, so it may fall in the fault region of Figure 2 and be misclassified as a fault. Conversely, a localized fault confined to a single sensor or a few weakly coupled sensors exhibits little correlation structure and a low anomalous-sensor ratio, so it may resemble an attack; this single-sensor ambiguity is most acute when correlation cues are minimal, since a localized fault and a targeted single-sensor attack can yield similar anomaly scores. The current criteria cannot fully separate these cases. Mitigations that preserve the interpretability of the approach—and that we leave to future work—include temporal pattern analysis, actuator-command-log integration, process-phase context, and physics-based domain rules that supplement the statistical correlation signal.

Dependence on the underlying detector. Because the categorization and interpretation layer operates on top of an upstream anomaly detector, the reported false-alarm identification precision (83% for false negatives and 73% for false positives) and the coverage of fault/attack separation inherit the quality of that detector. Pairing the framework with a stronger or differently designed detector would change these numbers and is worth studying separately.

Scope of evaluation. Two aspects fall outside the present scope. First, constructing per-time-point top-K correlation graphs and applying dynamic smoothing scales with the sensor count and sampling rate; at SWaT scale (51 attributes, second-level sampling) the cost

is moderate, but deployment on larger ICS with hundreds or thousands of sensors would require optimizing the graph-construction step. Second, the interpretations are assessed through case studies and visual artifacts—correlation heatmaps and impact subgraphs—rather than through quantitative interpretability metrics, the design of which for correlation-aware time-series XAI remains an open problem.

In conclusion, the discussion emphasized the role of the proposed framework as a post-detection categorization and interpretation layer for ICS anomaly analysis: it differentiates faults, attacks, and false alarms based on correlation characteristics, surfaces ambiguous decisions for expert review, and provides graph-based explanations that go beyond per sensor feature attribution. The limitations above delimit the scope of the present validation and motivate the future work outlined in the next section.

7. Conclusions and Future Work

This paper proposed a graph-based framework for the post-detection analysis of anomalies in industrial control systems. The framework separates faults from attacks using correlation characteristics, identifies candidate false positives and false negatives near the detection threshold through dynamic threshold smoothing, and produces correlation-aware graph explanations for each decision point. Validated on the SWaT dataset, the framework recovers the intended fault and attack categories, identifies false negatives with 83% precision and false positives with 73% precision, and yields impact-sensor subgraphs that match the documented propagation paths of the SWaT scenarios.

By replacing fixed-threshold blanket triage with characteristic-based categorization, the framework reduces the analytical burden on experts and provides a more credible basis for downstream review in availability-critical environments. The graph-based interpretation also generalizes beyond ICS: any multivariate time-series setting whose sensors exhibit physical or statistical coupling can benefit from explanations that expose those couplings, rather than attributing importance to features in isolation.

Future work, motivated by the limitations discussed in Section 6, includes cross-dataset validation on further testbeds such as WADI and HAI and on production ICS, automatic adaptation of the category thresholds to new ICS environments, integration of the framework with alternative anomaly detectors to study how its precision tracks detector quality, scalability optimization for larger sensor counts, and the development of quantitative interpretability metrics for correlation-aware XAI in time-series settings.

Author Contributions: Conceptualization, S.L. (Seonwoo Lee) and T.L.; methodology, S.L. (Seonwoo Lee); software, S.L. (Seonwoo Lee) and S.L. (Seungbeom Lim); validation, S.L. (Seonwoo Lee), S.L. (Seungbeom Lim) and T.L.; formal analysis, S.L. (Seonwoo Lee); investigation, S.L. (Seonwoo Lee); resources, T.L.; data curation, S.L. (Seonwoo Lee); writing—original draft preparation, S.L. (Seonwoo Lee); writing—review and editing, S.L. (Seungbeom Lim) and T.L.; visualization, S.L. (Seonwoo Lee); supervision, T.L.; project administration, T.L.; funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2026-25509873, Developing a Simulator for Cyber Defense Maturity Assessment and Autonomous Security). This work was also supported by the Gachon University research fund of 2024 (GCU-202500640001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The SWaT dataset used in this study is available upon request from iTrust, Singapore University of Technology and Design (SUTD).

Conflicts of Interest: Author Seungbeom Lim was employed by the company SANDS Lab. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
FN	False Negative
FP	False Positive
GAT	Graph Attention Network
GNN	Graph Neural Network
ICS	Industrial Control Systems
SWaT	Secure Water Treatment
TN	True Negative
TP	True Positive
XAI	Explainable Artificial Intelligence

References

1. Makridakis, S. The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms. *Futures* **2017**, *90*, 46–60. [CrossRef]
2. HaddadPajouh, H.; Dehghantanha, A.; Parizi, R.M.; Aledhari, M.; Karimipour, H. A survey on internet of things security: Requirements, challenges, and solutions. *Internet Things* **2021**, *14*, 100129. [CrossRef]
3. Sakhnini, J.; Karimipour, H.; Dehghantanha, A.; Parizi, R.M.; Srivastava, G. Security aspects of Internet of Things aided smart grids: A bibliometric survey. *Internet Things* **2021**, *14*, 100111. [CrossRef]
4. Karimipour, H.; Dinavahi, V. Robust massively parallel dynamic state estimation of power systems against cyber-attack. *IEEE Access* **2018**, *6*, 2984–2995. [CrossRef]
5. Lopez, F.; Saez, M.; Shao, Y.; Balta, E.C.; Moyne, J.; Mao, Z.M.; Barton, K.; Tilbury, D. Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1885–1892. [CrossRef]
6. Rojat, T.; Puget, R.; Filliat, D.; Del Ser, J.; Gelin, R.; Díaz-Rodríguez, N. Explainable Artificial Intelligence (XAI) on Time-Series Data: A Survey. *arXiv* **2021**, arXiv:2104.00950.
7. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
8. Song, Y.; Xin, R.; Chen, P.; Zhang, R.; Chen, J.; Zhao, Z. Identifying performance anomalies in fluctuating cloud environments: A robust correlative-GNN-based explainable approach. *Future Gener. Comput. Syst.* **2023**, *145*, 77–86. [CrossRef]
9. Urbina, D.I.; Giraldo, J.A.; Cardenas, A.A.; Tippenhauer, N.O.; Valente, J.; Faisal, M.; Ruths, J.; Candell, R.; Sandberg, H. Limiting the impact of stealthy attacks on industrial control systems. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1092–1105.
10. Marcílio, W.E.; Eler, D.M. From explanations to feature selection: Assessing SHAP values as feature selection mechanism. In Proceedings of the 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Porto de Galinhas, Brazil, 7–10 November 2020; pp. 340–347.
11. Tang, M.; Lau, V.K.N. Online Identification and Temperature Tracking Control for Furnace System With a Single Slab and a Single Heater Over the Wirelessly Connected IoT Controller. *IEEE Internet Things J.* **2024**, *11*, 6730–6747. [CrossRef]
12. Lalam, N.; Bhatta, H.; Bukka, S.R.; Zhang, P.; Diemler, N.; Shumski, A.; Ohodnicki, P.R.; Buric, M.P.; Wright, R.F. Deep Neural Network Assisted Distributed Strain and Temperature Fiber Sensor System for Natural Gas Pipeline Monitoring. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 7003711. [CrossRef]
13. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [CrossRef] [PubMed]
14. Lundberg, S.M.; Erion, G.G.; Lee, S.I. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv* **2018**, arXiv:1802.03888.

15. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef] [PubMed]
16. Schweidtmann, A.M.; Rittig, J.G.; Weber, J.M.; Grohe, M.; Dahmen, M.; Leonhard, K.; Mitsos, A. Physical pooling functions in graph neural networks for molecular property prediction. *Comput. Chem. Eng.* **2023**, *172*, 108202. [CrossRef]
17. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
18. Luan, S.; Hua, C.; Xu, M.; Lu, Q.; Zhu, J.; Chang, X.-W.; Fu, J.; Leskovec, J.; Precup, D. When Do Graph Neural Networks Help with Node Classification? Investigating the Impact of Homophily Principle on Node Distinguishability. In Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 28748–28760.
19. Filonov, P.; Lavrentyev, A.; Vorontsov, A. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-Based Predictive Data Model. *arXiv* **2016**, arXiv:1612.06676.
20. Ko, J.U.; Na, K.; Oh, J.S.; Kim, J.; Youn, B.D. A new auto-encoder-based dynamic threshold to reduce false alarm rate for anomaly detection of steam turbines. *Expert Syst. Appl.* **2022**, *189*, 116094. [CrossRef]
21. Qi, Y.; Shen, C.; Wang, D.; Shi, J.; Jiang, X.; Zhu, Z. Stacked Sparse Autoencoder-Based Deep Network for Fault Diagnosis of Rotating Machinery. *IEEE Access* **2017**, *5*, 15066–15079. [CrossRef]
22. Hwang, C.; Lee, T. E-SFD: Explainable sensor fault detection in the ICS anomaly detection system. *IEEE Access* **2021**, *9*, 140470–140486. [CrossRef]
23. Caville, E.; Lo, W.W.; Layeghy, S.; Portmann, M. Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. *Knowl.-Based Syst.* **2022**, *258*, 110030. [CrossRef]
24. Ali, S.; Abuhmed, T.; El-Sappagh, S.; Muhammad, K.; Alonso-Moral, J.M.; Confalonieri, R.; Guidotti, R.; Del Ser, J.; Díaz-Rodríguez, N.; Herrera, F. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Inf. Fusion* **2023**, *99*, 101805. [CrossRef]
25. Tripathy, S.M.; Chouhan, A.; Dix, M.; Kotriwala, A.; Klöpper, B.; Prabhune, A. Explaining Anomalies in Industrial Multivariate Time-Series Data with the Help of eXplainableAI. In Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp), Daegu, Republic of Korea, 17–20 January 2022; pp. 226–233.
26. Ye, L.; Keogh, E. Time series shapelets: A new primitive for data mining. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 947–956.
27. Sakurada, M.; Yairi, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, Gold Coast, Australia, 2 December 2014; pp. 4–11.
28. Deng, A.; Hooi, B. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 4027–4035.
29. Adepu, S.; Mathur, A. An Investigation into the Response of a Water Treatment System to Cyber Attacks. In Proceedings of the 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE), Orlando, FL, USA, 7–9 January 2016; pp. 141–148.
30. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security, CRITIS 2016*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10242, pp. 88–99.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.