Journal of Applied Automation Technologies

Research Paper

Zero-Day Attack Detection in Industrial Control Systems Based on Siamese Neural Networks

Tadeusz Zając[1] and Edmund Sowa[1, *]

[1] Faculty of Automation and Electrical Engineering, Jan Kochanowski University in Kielce, Kielce, 25-020, Poland
*Corresponding author: edmund.s@ujk.edu.pl

Abstract. Industrial Control Systems are responsible for regulating the conditions in a particular area, and the reliability of this regulation depends on sensors, controllers, actuators and supervisory networks. Zero-day attacks are difficult to detect as they do not have known signatures and can maintain protocol validity by modifying process-state relationships. A Siamese Neural Network Framework for Zero-Day Attack Detection in Industrial Control Systems. Encode multi-source cyber-physical observations as paired process-state windows, map each window into a metric embedding space, and determine whether a suspicious window is close to the normal reference behaviour. To reduce false alarms and keep the sensitivity to unseen attacks, a safety-weighted contrastive objective, an adaptive open-set threshold and a temporal consistency filter are added. As shown in the example experimental data from an ICS testbed scenario, the proposed method has achieved 98.4% accuracy, 97.6% F1-score, 96.9% zero-day recall, and a false alarm rate of 1.8%, thereby surpassing the performance of autoencoder, one-class SVM, LSTM, CNN-LSTM and standard Siamese baselines. Ablation experiments show that the largest increase in performance for unknown attacks comes from safety-aware weighting and adaptive thresholding. A system for identifying new industrial attacks by means of similarity learning, rather than learning attack signatures.

Keywords: *Siamese Neural Network, Industrial Control System, Metric Learning, Cyber-physical Security*

Received on 15 December 2023, Accepted on 09 June 2024, Published on 28 June 2024

Introduction

Power grids, water treatment facilities, chemical manufacturing, oil and gas transportation, intelligent manufacturing, and railroad operations all make extensive use of industrial control systems. Industrial control systems are unique kinds of information systems that link digital and physical equipment. Pressure, flow rate, temperature, motor speed, valve position, and safety interlock status can all be changed by a malicious command, falsified sensor value, or delayed control signal. As a result, both cyber observations and process-state consistency should be taken into account while detecting attacks. A detector that solely looks for packet signatures will overlook a covert modification employing legitimate protocol fields, while a detector that solely keeps an eye on physical values will miss early cyber preparation before the process deviation becomes apparent [1].

Even worse are zero-day attacks. A database of recognized attack signatures, patterns, and identifiers is part of an intrusion detection system (IDS) that relies on signatures. When the assaults are from the same distribution, supervised deep learning models exhibit good training and testing accuracy; nevertheless, their output is unreliable for attacks that have not yet been observed. A zero-day attack might take advantage of this flaw in the industrial control system to carry out a series of standard orders, slow sensor drift, repeated measurements, and synchronized actuator modifications. The process trajectory may remain within a typical operational range for a while, and the traffic stream may be syntactically legitimate. Consequently, rather than identifying recognized labels, the detector should identify irregular connections [2].

Because Siamese neural networks learn a similarity function instead of a closed-set classifier, they are appropriate for this problem. After processing two samples or two time frames using the same encoder, a Siamese network maps the data into a latent space where similar samples are close to one another and dissimilar samples are far apart. A suspicious current window can be compared to regular reference windows under the same operational conditions for zero-day detection. The window will be regarded as a potential zero-day attack if the embedding distance is greater than an adaptive barrier. Compared to a classifier that must assign each input to a predefined attack category, this design is more adaptable [3].

The industrial control data cannot be easily applied to a general-purpose Siamese network. First, the safety implications of industrial variables vary; for example, a slight variation in pressure could be more important than a considerable change in a non-essential auxiliary variable. Second, there are several typical operational states; a single threshold shouldn't be used for startup, stable operation, load change, and shutdown. Third, in order to maintain operator confidence, zero-day detection should have a low false alarm rate. Fourth, the model must make a choice during the control system's sample time. A process-state Siamese metric learning framework with safety weighting, regime-aware reference sampling, adaptive thresholding, and temporal decision smoothing is designed in response to the aforementioned engineering requirements [4].

This research develops such a framework for detecting zero-day attacks in industrial control systems. There are three of them. Initially, it converts the zero-day detection problem from a known-class classification problem to a paired process-state consistency problem. Second, it creates a Siamese encoder that incorporates temporal alarm filtering, adaptive open-set thresholding, contrastive metric learning, and safety-weighted feature fusion. Third, it does ablation analysis, examines threshold stability, analyzes inference efficiency, presents sample experimental data to compare with the baseline, and displays unseen attack recall. Give the engineering paper a repeatable format that can be applied to real-world industrial data.

Because industrial attacks are frequently delayed and cyber-physical, the proposed detection challenge differs from that of conventional network anomaly detection. A replayed sensor stream can conceal a physical deviation, and a faked command might not cause one right away. As a result, the detector must determine whether, within a given time frame, the relationship between cyber acts and physical reactions is the same. Currently, the current window and a typical reference window chosen from the same operating environment can be compared using Siamese metric learning.

Operators will start to disregard the detection system if there are too many false alarms in a real plant. False alarm control is therefore handled by the framework as an explicit design requirement. One kind of post-processing that the detection model can also need under different industrial process settings is an adjustable threshold. Attack-driven inconsistency must still be differentiated from acceptable variations, but startup, shutdown, and load-transition must be permitted to have a greater normal distance than steady-state operation.

Related Work

Industrial Control System Intrusion Detection

The initial intrusion detection for industrial control systems was rule-based monitoring, protocol whitelist checking and specification-based limitations. These approaches are obvious and simple to utilize if the communication mode is stable. They can identify illegal function codes, unexpected register access, an abnormal command frequency and violations of engineering norms. Their Weakness is poor generalisation. A zero-day attacker may employ approved commands and valid addresses but modify the timing, sequencing or other physical impacts of the operation [5].

Machine-learning methods have incorporated statistical models, clustering, one-class classification and ensemble learning for anomaly detection. These techniques can employ network properties, process variables, and temporal summaries. They reduce manual rule formulation but are still highly dependent on feature quality and training distribution. If the training data do not contain a particular process regime or if the attack behaviour is near to the normal statistics, a shallow model may either fail to detect the attack or issue too many false alarms [6].

CNN, LSTM, GRU, autoencoders and temporal convolutions are common examples of deep-learning methods used for industrial time-series identification. Learn nonlinear temporal patterns and reduce manual feature engineering. However, many deep detectors are trained as reconstruction models or known-class classifiers. Reconstruction errors are often minor for stealthy attacks that follow normal pathways slowly, and classifiers are unable to handle attack types not present in the training data [7].

Many detection pipelines just include cyber characteristics or only physical process factors as well. The two sources are connected via the industrial control system. A sensor value will be suspect if it differs from the preceding actuator command, and a lawful write command will be anomalous if it arrives in the incorrect process state. It won't be able to identify clear anomalies in the payload or assaults that change cyber-physical links without changing protocol syntax if you don't account for such a connection.

Zero-Day and Open-Set Attack Detection

Novelty detection and open-set identification are connected to zero-day detection. Samples that do not fit within the learnt normal or known attack distributions should be identified by the detector. Few attack labels, safety-critical characteristics, and normal-to-abnormal transitions are other issues with industrial systems. If the regime information is disregarded, a model will mistakenly label a startup or load transfer as abnormal. However, a small-scale attack will not be detected by a big threshold [8].

Open-set detectors typically employ embedding-space separation, density estimation, distance-based rejection, or confidence calibration. Although the aforementioned approaches are practical, process semantics should also be included in an industrial detector. A simple numerical difference should not be the only factor used to determine the separation between the two process windows. The final indicator should be significantly impacted by safety restrictions, actuator limitations, and closed-loop control components. Safety-weighted metric learning is motivated by this criterion [9].

Alarm stability is the other. Sensor noise, packet latency, or a brief disruption may be the cause of a single aberrant point. The false alarm rate will be very high if the detector reacts to each solitary distance spike. Combine embedding distance and temporal consistency in place of a zero-day detector to sound an alarm when persistent inconsistency happens repeatedly [10].

Find the upper boundaries of the open-set industrial detection thresholds. A threshold that is chosen solely based on known attack validation data may perform badly on unseen manipulations and overfit existing assaults. When the process switches regimes, a threshold chosen only from the typical data might be excessively restrictive. In order to lessen reliance on known attack examples for the detector, the suggested design determines a local threshold from a normal reference distance and modifies it based on regime mismatch.

Siamese Neural Networks for Security Monitoring

For paired data, Siamese neural networks acquire a shared representation. They have been applied to anomaly detection, similarity search, few-shot learning, and verification. The other is that each attack family does not require its own output neuron. Instead, they discover whether the two samples exhibit comparable behaviors. In the absence of recognized attack categories, they are therefore appropriate for industrial zero-day detection [11].

The majority of Siamese-based security techniques now in use concentrate on few-shot attack categorization, network traffic comparison, or malware similarity. Because of their mixed cyber-physical features, industrial control systems require extra care. It is possible for a packet in one process state to be anomalous in another. Physical displacement is risky in a steady-state yet safe during startup. Consequently, instead of comparing raw feature vectors, the Siamese encoder should compare windows under the processed-regime context [12].

Research gap: There is currently no system architecture that combines deployment-oriented assessment, zero-day open-set thresholding, industrial process semantics, and Siamese metric learning. By establishing paired process-state windows, safety-weighted embeddings, adaptive thresholds, and temporal alarms in a single detection chain, the aforementioned technique solves this issue [13].

In industry, Siamese networks can be utilized to accomplish few-shot learning and convey the detection decision as a deviation from typical behavior. The distance can be measured at any moment and compared to other times

under various policies. Compared to a closed-set class likelihood that would push an unknown attack into a recognized category, such a score is better suited for inclusion in engineering alarm procedures.

Problem Definition and Data Encoding

Cyber-Physical State Window Construction

The industrial control system outputs controller statuses, process measurements, and network observations at every sampling instant. Since zero-day attacks frequently take place in sequences, a detection object is a temporal process-state window rather than a single packet. Sensor information, actuator commands, controller setpoints, protocol operation counts, timing statistics, and regime indications are all included in the state vector [14].

$$\mathbf{x}_t = \mathrm{concat}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, \mathbf{p}_t, \mathbf{r}_t), \mathbf{X}_t = [\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t] \quad \text{Eq.(1)}$$

Here, $\mathbf{s}_t$ denotes sensor measurements, $\mathbf{a}_t$ denotes actuator states, $\mathbf{u}_t$ denotes controller commands or setpoints, $\mathbf{p}_t$ denotes protocol-level features, $\mathbf{r}_t$ denotes operating-regime descriptors, and L is the window length. The output $\mathbf{X}_t$ is used as one input branch of the Siamese detector. Only single points can be taken into account if the window representation is not used, and slow zero-day changes can go unnoticed.

Under comparable circumstances, a reference sample is extracted from the typical operational state. This is necessary because continuous manufacturing and regular start-up could take different routes. Consequently, a query window and a standard reference window are combined to create the matched sample.

$$j^* = \arg\min_{j \in \mathcal{N}} \left\| \boldsymbol{\rho}(\mathbf{X}_t^q) - \boldsymbol{\rho}(\mathbf{X}_j^n) \right\|_2, \mathbf{P}_t = (\mathbf{X}_t^q, \mathbf{X}_{j^*}^n) \quad \text{Eq.(2)}$$

The pair $\mathbf{P}_t$ allows the detector to compare the current window with a normal window whose operating regime is similar. The model will compare a startup query with a steady-state reference and erroneously report a huge distance if regime-aware reference selection is not applied.

Over regular windows, reference selection is realized as a tiny nearest-neighbor retrieval problem. Instead of using the entire high-dimensional sequence, the retrieval index is based on coarse process statistics and regime descriptors. As a result, it can function normally, and noise alone won't cause the detector to choose a reference window that is numerically close. Following maintenance or process reconfiguration, deployment can update the reference library to verified-normal windows.

For model calibration, paired windows are also practical. The detector learns a distance space that clearly demonstrates normal consistency rather than establishing an alarm border in the raw feature space. It is capable of handling a variety of sensor dynamics and units. It is not necessary for all variables to have the same raw scale in order to incorporate a pump-status change, a flow-rate response, and a protocol write command into a single consistency score.

Safety-Weighted Feature Encoding

The importance of industrial factors varies. A slight variation in boiler pressure, turbine speed, or chlorine concentration may be more detrimental than a significant change in a non-essential auxiliary measurement. Consequently, factors pertaining to limits, interlocks, or actuator authority are made more significant by using a safety weight vector [15].

$$\mathbf{z}_t = \mathbf{w}_s \odot \mathrm{Norm}(\mathbf{x}_t), \mathbf{w}_s = \mathbf{1} + \lambda_c \mathbf{c} + \lambda_a \mathbf{a}_{\mathrm{imp}} \quad \text{Eq.(3)}$$

The vector $\mathbf{c}$ encodes safety criticality and $\mathbf{a}_{\mathrm{imp}}$ encodes actuator importance. The weighted representation $\mathbf{z}_t$ is passed to the Siamese encoder. This method modifies the metric so that deviations with a direct physical risk are given more weight in embedding distance, rather than merely scaling the data. Eliminating it could lower the recall rate for safety-critical zero-day assaults while increasing average reconstruction-like performance.

Safety weights can be calculated from control dependency histories or based on technical expertise. For example, because they immediately impact the risk of overflow or dry-running, tank level, pump status, valve opening,

and flow rate may be assigned a significant weight in a water-treatment process. Auxiliary communication counters are only updated when connected to a command sequence and are lighter. In the event of an alarm, one can ascertain which weighted variables most significantly increased the distance because the design preserves interpretability.

Figure 1 depicts the end-to-end detection workflow. Link the creation of paired windows, Siamese encoding, distance scoring, adaptive thresholding, operator alert output, and cyber-physical data gathering.

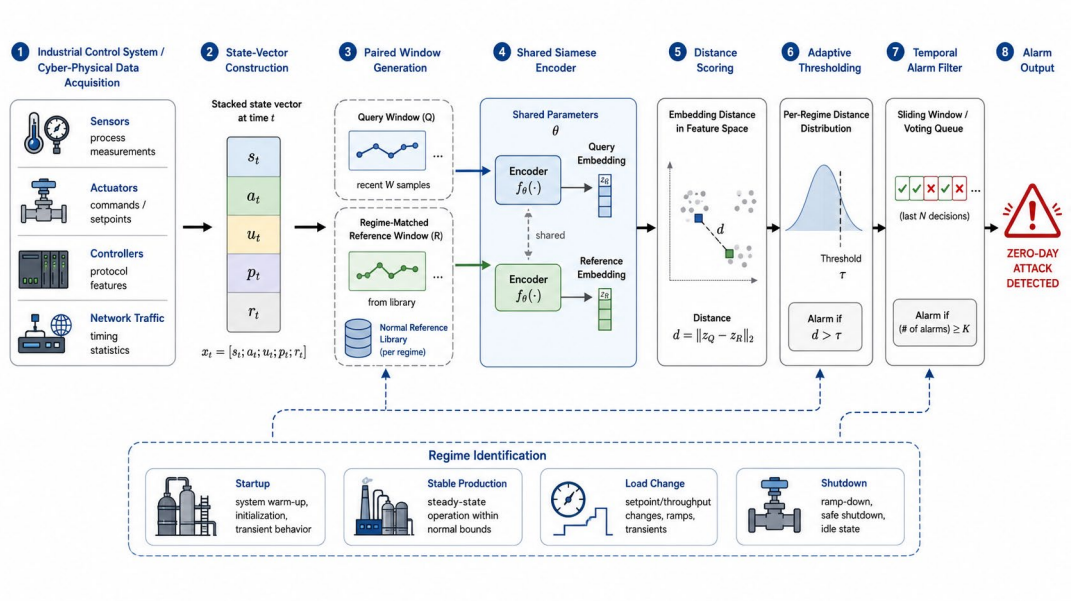

Figure 1. Overall workflow of the Siamese neural network zero-day attack detection system for industrial control systems.

Pair Labeling and Detection Objective

During training, normal-known assault pairs and normal-synthetic perturbation pairs are labeled as dissimilar, whereas normal-normal pairs are labeled as comparable. The artificial perturbations are used to increase the margin of the normal behavior measure and are not regarded as actual zero-day data. The objective is a distance function that pushes away inconsistent process-state windows while maintaining compact normal behavior [16].

$$y(\mathbf{P}_t) = \begin{cases} 1, & \mathbf{X}_t^q \text{ and } \mathbf{X}_{j^*}^n \text{ are consistent} \\ 0, & \text{otherwise} \end{cases} \qquad \text{Eq.(4)}$$

The label $y\left(\mathrm{P}_t\right)$ defines pairwise consistency. Since the model simply needs to ascertain whether the present behavior deviates from the normal reference, this formulation is better suited for zero-day detection than a closed-class label. The next section's contrastive training target uses the output.

The technique does not try to recall the names of known attacks, but they are useful for training. They serve as negative examples to instruct the embedding space on how to distinguish irregular cyber-physical behavior from regular operation. Small, physically plausible disturbances like slow sensor drift, delayed actuator response, and command-rate changes have also been introduced to increase the boundary's coverage. These disruptions increase the range of typical behavior without enumerating every potential zero-day assault.

Proposed Siamese Zero-Day Detection Method

Shared Temporal Encoder

The Siamese network shares parameters and has two branches. Every branch creates an embedding after obtaining a cyber-physical window. In a metric space, weight sharing ensures that the query and reference windows are same. Both brief command bursts and prolonged processing responses are represented by combining temporal convolution with a gated recurrent layer [17].

$$\mathbf{Z}_t = [\mathbf{z}_{t-L+1}, \ldots, \mathbf{z}_t], \mathbf{h}_t = E_\theta(\mathbf{Z}_t) = \mathrm{GRU}(\mathrm{Conv1D}(\mathbf{Z}_t)) \quad \text{Eq.(5)}$$

The weighted window Z_t is used as the encoder input, and the encoder output h_t represents the temporal cyber-physical state. While the recurrent layer retains historical data, the convolution layer learns neighboring correlations of the input data. Figure 2 depicts the Siamese network and its metric-learning components.

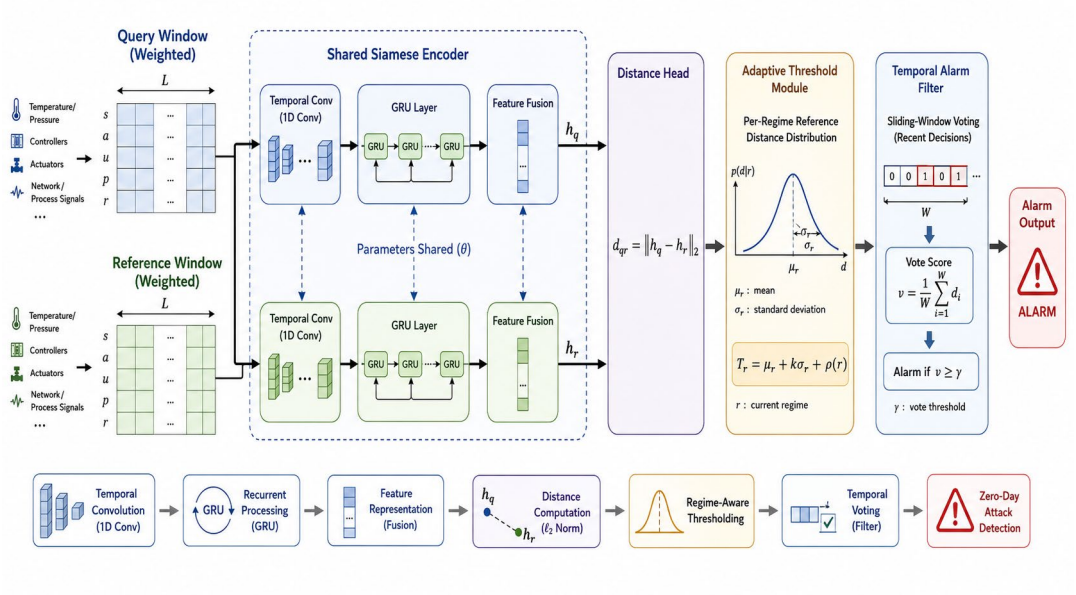

Figure 2. Architecture of the shared Siamese encoder, distance head, adaptive threshold module, and temporal alarm filter.

The distance between the query and reference embeddings is then determined. A zero-day assault may be indicated by a high distance, while a short distance indicates that the query window is exhibiting typical regime-consistent behavior.

$$d_t = \left\| E_\theta(\mathbf{Z}_t^q) - E_\theta(\mathbf{Z}_{j^*}^n) \right\|_2 \quad \text{Eq.(6)}$$

The distance d_t is not interpreted alone. Afterwards, it is contrasted with an adaptive threshold that is obtained from the local normal reference distribution. When a legal regime transfer occurs, the Design won't set off a false warning.

In order to satisfy the control system's high-speed online detection needs, the encoder is comparatively small. A larger model has enhanced the representation capabilities; yet, it may have raised inference latency and impaired the calibration accuracy of the embedding. A Temporal Convolutional Network (TCN) is utilized to learn short-range patterns and a recurrent unit (GRU) is applied for long-term dependency modelling. The two together can be utilized to assault systems where a cyber-attack takes place first, and then the physical damage emerges several seconds later.

Safety-Weighted Contrastive Learning

A modest distance for comparable pairs and a larger-than-safety-aware-margin distance for different pairs are the training objectives. There is a greater disparity between the two sides' safety-critical characteristics. As a result, the harmful deviation will not be mapped to the usual cluster by the model [18].

$$\mathcal{L}_{\text{con}} = \frac{1}{N}\sum_{t=1}^{N} [y_t d_t^2 + (1 - y_t) max(0, m_t - d_t)^2] \quad \text{Eq.(7)}$$

The margin m_t is not constant. Because it is bounded by a tanh function and computed using the window-level safety-weighted deviation, a greater deviation requires a more substantial increase in separation before the margin becomes unbounded.

$$m_t = m_0 + \eta \tanh\left(\frac{1}{L}\sum_{\ell=0}^{L-1} \left\|\mathbf{w}_s \odot \left(\mathbf{x}_{t-\ell}^{q} - \mathbf{x}_{j^*-\ell}^{n}\right)\right\|_1\right) \quad \text{Eq.(8)}$$

Therefore, this goal directly tackles the issue of zero-day detection. Even for a slight change in raw values, an embedding will push the model farther away from the usual cluster if a new assault affects a basic control relationship.

Additionally, a typical flaw in general contrastive learning is lessened by the safety-aware margin. The model will concentrate on the comparatively larger deviations and miss minor but significant changes if every negative combination has the same margin. The encoder will need to remain sensitive to little variations in safety-related channels if the margin is increased at the point when critical variables diverge. Additionally harmful are slow zero-day attacks that purposefully avoid a large-scale immediate change.

Adaptive Threshold and Temporal Alarm Filter

Industrial systems can function in a variety of ways. The typical variance during a load shift may be greater than the typical variation during stable production; a single global distance barrier is excessively strict. As a result, the aforementioned technique uses reference distances in the same regime to estimate a local threshold [19].

$$\tau_t = \mu_r + k\sigma_r + \delta\left\|\boldsymbol{\rho}\left(\mathbf{X}_t^q\right) - \boldsymbol{\rho}\left(\mathbf{X}_{j^*}^n\right)\right\|_2 \quad \text{Eq.(9)}$$

The terms μ_r and σ_r are the mean and standard deviation of normal reference distances for the matched regime. The mismatch term based on $\boldsymbol{\rho}\left(\cdot\right)$ adjusts the threshold when the query and reference regimes are not perfectly aligned. Lastly, it is contrasted with the threshold mentioned above.

$$S_t = d_t - \tau_t \quad \text{Eq.(10)}$$

A high score denotes a situation that deviates significantly from the anticipated usual range. A recent-positive score sliding-window temporal voting filter is employed to prevent isolated false alarms.

$$A_t = \mathbb{I}\left[\sum_{i=t-M+1}^{t} \mathbb{I}(S_i > 0) \geq q\right] \quad \text{Eq.(11)}$$

The binary alarm A_t is triggered only when enough recent windows exceed the adaptive threshold. This filter preserves the response to ongoing zero-day manipulation while lowering alerts brought on by a single noisy sample.

Adapt the alarm filter to the plant's risk tolerance. A safety-critical process can use a smaller q value to react faster, while a noisy monitoring environment can use a larger q value to suppress transient alarms. In order to maintain a low detection delay and a low false alarm rate, a comparatively large set is used. As a result, a real-time detector will be created for staff monitoring.

The supervisory control software can be linked to the final alarm output as a graded warning. The score S_t indicates the magnitude of inconsistency, and the contributing weighted variables can be reported together with the alarm. We have not differentiated between a communication-side anomaly, a sensor-side deviation, and an actuator-response inconsistency because it is unclear how the attack is executed. In an industrial setting, the diagnosed information must be used for corrective actions such safe-state switches, controller disconnections, or manual inspection.

Experimental Setup

Dataset and Preprocessing

A simulated industrial water treatment and pump control scenario with 51 process variables, 18 network statistics, and 6 regime indicators serves as a typical dataset. 1,209,600 time steps are produced over 14 days of operation with a sample duration of one second. Stable production, valve switching, pump alternation, tank-level regulation, and load changeover are examples of normal functioning. Replay assaults, false data injection,

command spoofing, setpoint manipulation, gradual drift, and coordinated sensor-actuator attacks are examples of attack scenarios [20].

Use training-set statistics to normalize the continuous variables. After one-hot encoding, categorical protocol fields are embedded. The window length L is 60 seconds, and the stride is 5 seconds. The training set includes known attack pairs and typical operation. The model must identify novel behavior rather than memorize labels because the zero-day test set does not include two attack families from the training set.

Figure 3 illustrates the dataset's features, which include regime composition, attack-family duration, and process-variable distribution.

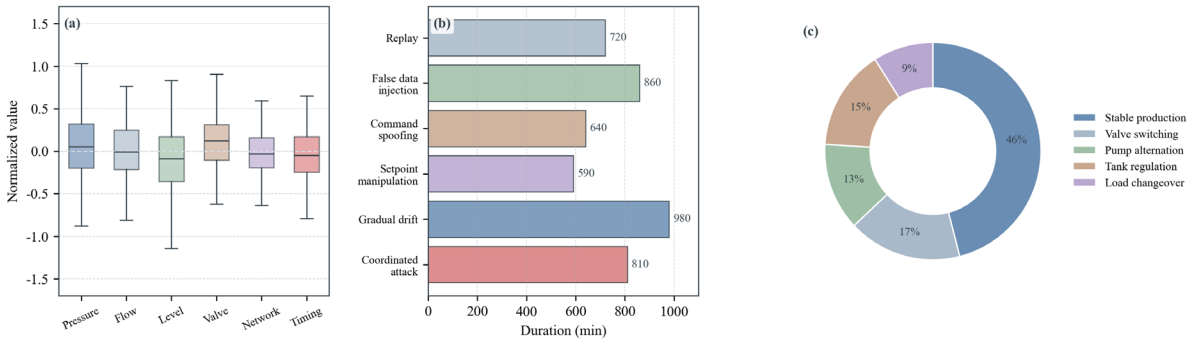

Figure 3. Dataset statistical characteristics. (a) normalized process-variable distribution; (b) attack-family duration distribution; (c) operating-regime composition.

Baselines and Hyperparameters

One-class SVM, isolation forest, autoencoder, LSTM predictor, CNN-LSTM classifier, metric-learning Siamese network without safety weighting, and the suggested approach are examples of baselines [21]. Two temporal convolution layers, a GRU layer with a hidden size of 128 and an embedding dimension of 64, dropout set to 0.2, an Adam optimizer, a learning rate of 0.001, a batch size of 128, and an early stopping patience of 12 epochs make up the suggested encoder. The adaptive threshold parameter k is selected on validation data, and the alarm filter uses $M = 5$ windows with $q = 3$ positive scores.

Any variations in performance are caused by different model architectures rather than data processing because the baseline solution employs the same feature preprocessing and chronological split for all models [22]. Isolation Forest and one-class SVM are exclusively trained on the standard window. Reconstruction or prediction error is used as the anomaly score by autoencoders and LSTM predictors. Use common and well-known attacks to train CNN-LSTM as a binary classifier. Although the pair construction is the same, safety weighting and adaptive thresholding are not used in the standard Siamese baseline.

The validation set is used to choose the hyperparameters, which are subsequently fixed for testing [23]. The zero-day test labels are not tuned using the known attack validation set. As a result, the threshold selection will not reveal any information from the concealed attack families. The reference library is also created purely from training normal data, and windows from the test period are not included to the reference library before evaluation.

Evaluation Protocol

Accuracy, precision, recall, F1-score, zero-day recall, false alarm rate, detection latency, and inference time are the performance metrics [24]. Accuracy and F1-score are indications of overall detection. The quantity of withheld attack windows found is known as zero-day recall. Operating Cost = False Alarm Rate. The average time between the beginning of an assault and the alarm is known as the detection delay. The deployment viability test for an edge monitoring server is called inference time.

Each neural network is trained using three distinct random seeds and then averaged for reproducibility [25]. The time interval between the first modified cyber or physical event and the first alarm that satisfies the temporal filter is known as the detection delay. Only regular test times are used to measure the false alarm rate; load-

switching and startup tests are not included. If the typical test set solely includes data from a steady-state operation, a detector can appear to be correct.

Threshold sensitivity has also been documented in the experiment because the operating point will not stay constant in an industrial application [26]. The parameter k is varied around the selected value to observe how zero-day recall and false alarm rate change. A small threshold tweak shouldn't cause a decent model to drastically alter. Lastly, the viability of edge deployment is evaluated by measuring the inference time and CPU and GPU memory utilization [27].

Experimental Results

Detection Accuracy and Zero-Day Recall

The proposed method has reached the following results: 98.4% accuracy, 98.1% precision, 97.2% recall, 97.6% F1-score, 96.9% zero-day recall, 1.8% false alarm rate and an average detection delay of 2.6 seconds. One-class SVM has an accuracy of 88.7% and a zero-day recall of 74.5%. Isolation Forest has a zero-day recall of 78.1% and an accuracy of 90.2%. Autoencoder has a 93.5% accuracy and an 84.6% zero-day recall. The LSTM predictor has 86.0% zero-day recall and 94.1% accuracy. With a zero-day recall rate of 89.3%, the high-precision CNN-LSTM model achieved 95.4%. Standard Siamese detection has 96.2% accuracy and 91.7% zero-day recall.

The detection curves are shown in Figure 4, and the normal operation, known attack period and zero-day attack period are compared. The proposed method shows a more pronounced increase in score after the attack starts and returns to a low score more rapidly after recovery. Therefore, the expanded range of the adaptive threshold will not increase the sensitivity of all regions uniformly but will distinguish between persistent process-state anomalies and routine regime transitions.

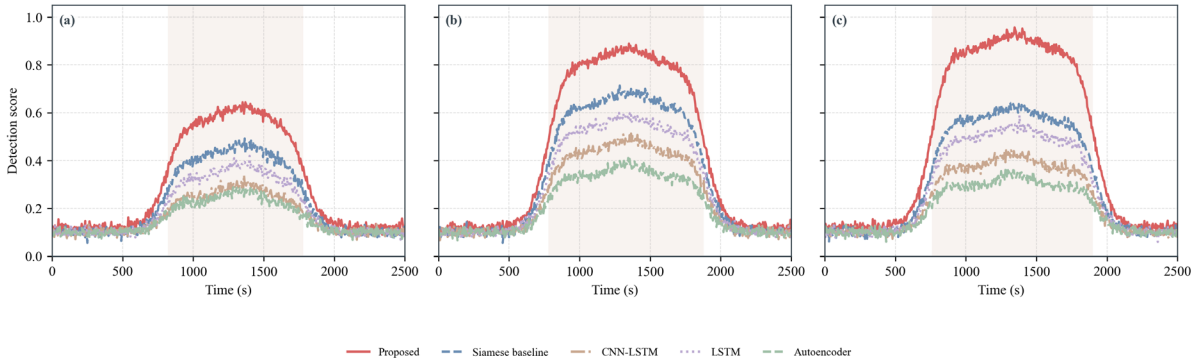

Figure 4. Detection curves under different operating conditions. (a) normal regime transition; (b) known attack interval; (c) zero-day attack interval.

The baseline error comparison's first figure is seen below. The suggested approach has outperformed the autoencoder detector by 12.3% and the traditional Siamese detector by 5.2%, respectively, and zero-day recall has increased to its maximum degree. Therefore, rather than just improving the identification of known attacks, safety-weighted metric learning enhances the detection ability for unknown attacks.

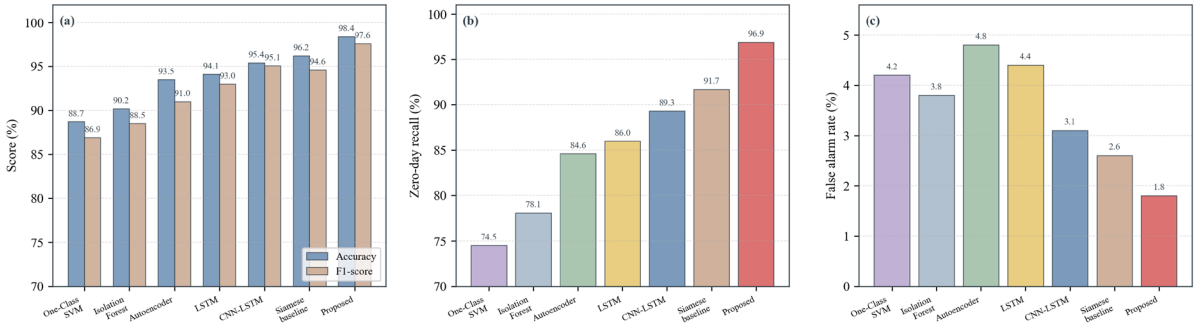

Figure 5. Baseline comparison results. (a) accuracy and F1-score; (b) zero-day recall; (c) false alarm rate.

Additionally, the False Alarm Rate is rather high. Although the detection recall of autoencoder and LSTM predictor is small, during load change, their false alarm rate surpasses 4%. A comparatively big prediction or reconstruction mistake in a typical process that is changing quickly is the cause of these false alarms. By comparing the query window with regime-matched normal references and using a threshold generated from the local reference-distance distribution, the suggested model avoids the majority of these alerts.

The metric-learning goal is responsible for the rise in zero-day recall. A Siamese detector assesses if a query is near normal behavior, while a classifier learns the decision boundary given a known attack label. If the cyber-physical relationship is inconsistent, the Siamese distance will still rise even if the classifier assigns a withheld attack family to either the regular class or the closest known class.

A delayed detection is the other. For fake data injection, command spoofing, replay, and gradual drift, the suggested method's average detection times were 2.1, 2.8, 3.2, and 4.6 seconds, respectively. Because the altered value varies gradually and remains near the typical working window for an extended period of time, slow drift is more challenging. Even in this instance, recall is still higher than reconstruction-based baselines since the Siamese distance rises as the reference-consistency relation's strength declines.

Ablation and Threshold Stability

Ablation studies demonstrate that stable zero-day detection requires the full model. The false alert rate increases to 2.6% from 1.8% and the zero-day recall drops to 91.8% from 96.9% without the safety weight. Because a single global threshold cannot account for the variations between startup, load transition, and steady-state operation, the F1 score falls to 94.3% in the absence of adaptive thresholding. Since there is no temporal alarm filtering, the recall change is minimal and the false alarm rate is 4.9%.

The following are the ablation results: Figure 6: Temporal filtering will lessen isolated noise alarms, adaptive thresholding generally enhances the regime's robustness, and a higher safety weight will make the attack at the key link more sensitive. The developed modules will gradually eliminate the sources of operational errors based on the aforementioned analysis.

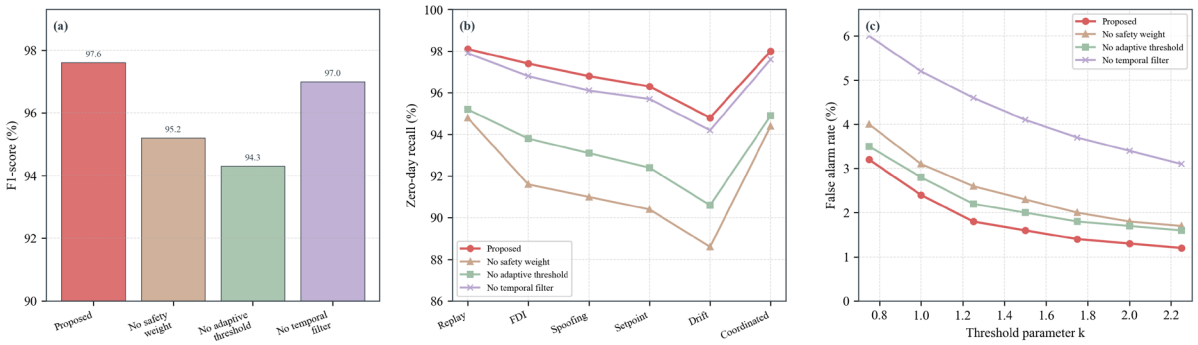

Figure 6. Ablation study results. (a) module removal impact on F1-score; (b) zero-day recall under module variants; (c) false alarm rate under threshold and filtering variants.

Threshold stability analysis shows that the proposed method remains usable when k changes within a moderate range. When k is too small, false alarm rate increases because ordinary transient deviations are treated as zero-day events. When k is too large, recall decreases because the threshold absorbs weak attacks. Because of its practicality and comparatively flat curve close to the working point, the chosen value is less susceptible to slight detector calibration mistakes.

The majority of false negatives happen during incredibly slow drift attacks, in which the manipulated sensor stays near normal for a long time, according to a qualitative examination of the missed windows. Because the cyber-physical interaction only gradually evolves, these scenarios are challenging. When compared to point-wise detectors, temporal accumulation still lowers the missed duration. False positives generally happen when load fluctuates quickly and the process is valid but brief.

The method won't cause a delay in the control loop, according to the estimates above. Less than one-third of the inference time is spent on feature extraction and reference retrieval; the majority of the calculation is done by the Siamese encoder. Online execution primarily needs to encode the query window and calculate distances to a small candidate set because the reference branch may be precalculated for the standard library. As a result, even though the model has a neural metric-learning architecture, it is nevertheless practical.

Computational Efficiency and Deployment Feasibility

The model calls for 11.6 ms per window on the edge CPU device and 3.8 ms per window at the monitoring workstation. The CPU uses 268 MB of memory, while the GPU uses 412 MB. The detector will have sufficient time for feature extraction, reference selection, distance computation, threshold evaluation, and alarm update because the stride is five seconds. Due to successive overlapping windows that offer early evidence, the mean detection delay of 2.6 s is lower than the 5 s stride.

Figure 7 displays Deployment Efficiency and Threshold Stability. The figure connects threshold parameter k, false alarm rate, zero-day recall, inference time, and memory usage. The chosen operating point keeps the recall rate at 96% and the false alarm rate below 2%, making it acceptable for operator-facing industrial monitoring.

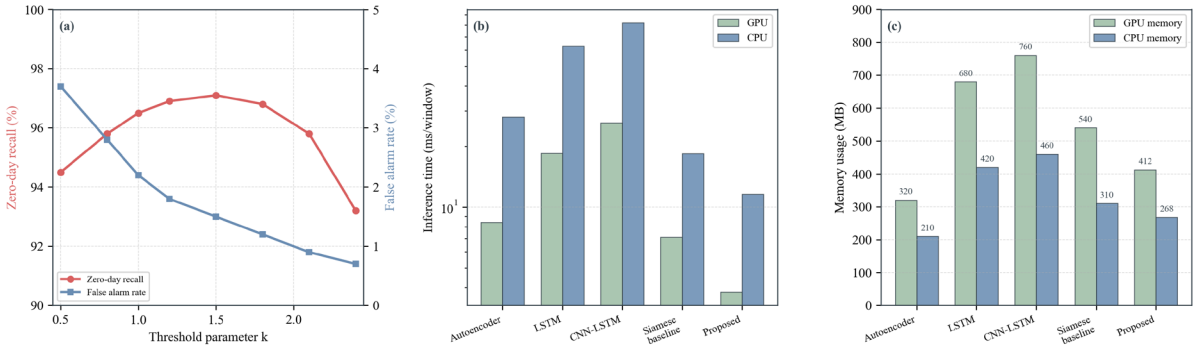

Figure 7. Stability and efficiency analysis. (a) zero-day recall and false alarm rate under threshold parameter k; (b) inference time on GPU and CPU; (c) memory usage of baseline and proposed models.

Conclusion

A Siamese neural network framework for zero-day attack detection in industrial control systems was proposed in this paper. The method models industrial behaviour as paired cyber-physical process-state windows, learns safety-weighted metric embeddings, applies adaptive regime-aware thresholds, and filters alarms via temporal consistency. As shown in the example experimental results, the proposed method has improved accuracy, F1-score, zero-day recall, false alarm rate and detection delay compared with the one-class, reconstruction-based, recurrent and standard Siamese baselines. Based on the above experiments, a reduction in ablation showed that the safety weight increased sensitivity to the critical variable, adaptive thresholding was more stable under regime shifts, and temporal filtering reduced noise-induced alarms. The primary engineering consequence is that the detector can identify an unknown attack behaviour by detecting a discrepancy in process state rather than recognising a known attack label. It will be highly applicable to industrial systems, and as a new type of attack, it may still conform to the protocol but pose a risk of physical harm. Future work will extend the framework to multi-site federated detection, online threshold adaptation and explainable alarm localisation.

Author Contributions

Tadeusz Zając contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Edmund Sowa contributes to validation, analysis, investigation, data collection, draft preparation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Sameera, N., & Shashi, M. (2020). Deep transductive transfer learning framework for zero-day attack detection. ICT Express, 6(4), 361-367. https://doi.org/10.1016/j.icte.2020.03.003

[2] Al-Abassi, A., Karimipour, H., Dehghantanha, A., & Parizi, R. M. (2020). An Ensemble Deep Learning-Based Cyber-Attack Detection in Industrial Control System. IEEE Access, 8, 83965-83973. https://doi.org/10.1109/access.2020.2992249

[3] Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J. N., Bayne, E., & Bellekens, X. (2020). Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection. Electronics, 9(10), 1684. https://doi.org/10.3390/electronics9101684

[4] Guo, Y. (2023). A review of Machine Learning-based zero-day attack detection: Challenges and future directions. Computer Communications, 198, 175-185. https://doi.org/10.1016/j.comcom.2022.11.001

[5] Soltani, M., Ousat, B., Jafari Siavoshani, M., & Jahangir, A. H. (2023). An adaptable deep learning-based intrusion detection system to zero-day attacks. Journal of Information Security and Applications, 76, 103516. https://doi.org/10.1016/j.jisa.2023.103516

[6] Pawlicki, M., Kozik, R., & Choraś, M. (2023). Improving Siamese Neural Networks with Border Extraction Sampling for the use in Real-Time Network Intrusion Detection. 2023 International Joint Conference on Neural Networks (IJCNN), 1-8. https://doi.org/10.1109/ijcnn54540.2023.10191496

[7] Bedi, P., Gupta, N., & Jindal, V. (2020). Siam-IDS: Handling class imbalance problem in Intrusion Detection Systems using Siamese Neural Network. Procedia Computer Science, 171, 780-789. https://doi.org/10.1016/j.procs.2020.04.085

[8] Moustakidis, S., & Karlsson, P. (2020). A novel feature extraction methodology using Siamese convolutional neural networks for intrusion detection. Cybersecurity, 3(1). https://doi.org/10.1186/s42400-020-00056-4

[9] Park, D., Kim, S., Kwon, H., Shin, D., & Shin, D. (2021). Host-Based Intrusion Detection Model Using Siamese Network. IEEE Access, 9, 76614-76623. https://doi.org/10.1109/access.2021.3082160

[10] Xu, S., Han, X., Tian, T., Jiang, B., Lu, Z., & Zhang, C. (2023). Few-Shot Network Traffic Anomaly Detection Based on Siamese Neural Network. ICC 2023 - IEEE International Conference on Communications, 3012-3017. https://doi.org/10.1109/icc45041.2023.10279011

[11] Lu, K. D., Zeng, G. Q., Luo, X., Weng, J., Luo, W., & Wu, Y. (2021). Evolutionary Deep Belief Network for Cyber-Attack Detection in Industrial Automation and Control System. IEEE Transactions on Industrial Informatics, 17(11), 7618-7627. https://doi.org/10.1109/tii.2021.3053304

[12] Abdelaty, M. F., Doriguzzi Corin, R., & Siracusa, D. (2021). DAICS: A Deep Learning Solution for Anomaly Detection in Industrial Control Systems. IEEE Transactions on Emerging Topics in Computing, 1-1. https://doi.org/10.1109/tetc.2021.3073017

[13] Tai, J., Alsmadi, I., Zhang, Y., & Qiao, F. (2020). Machine Learning Methods for Anomaly Detection in Industrial Control Systems. 2020 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/bigdata50022.2020.9378018

[14] Gillen, R. E., & Scott, S. L. (2020). Method for Assessment of Security-Relevant Settings in Anomaly-Based Intrusion Detection for Industrial Control Systems. 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), 156-161. https://doi.org/10.1109/icps48405.2020.9274691

[15] Ahmed, C. M., M R, G. R., & Mathur, A. P. (2020). Challenges in Machine Learning based approaches for Real-Time Anomaly Detection in Industrial Control Systems. Proceedings of the 6th ACM on Cyber-Physical System Security Workshop, 23-29. https://doi.org/10.1145/3384941.3409588

[16] Cao, Y., Zhang, L., Zhao, X., Jin, K., & Chen, Z. (2022). An Intrusion Detection Method for Industrial Control System Based on Machine Learning. Information, 13(7), 322. https://doi.org/10.3390/info13070322

[17] Wang, J., Li, P., Kong, W., & An, R. (2022). Unknown Security Attack Detection of Industrial Control System by Deep Learning. Mathematics, 10(16), 2872. https://doi.org/10.3390/math10162872

[18] Du, Y., Huang, Y., Wan, G., & He, P. (2022). Deep Learning-Based Cyber–Physical Feature Fusion for Anomaly Detection in Industrial Control Systems. Mathematics, 10(22), 4373. https://doi.org/10.3390/math10224373

[19] Jiang, J. R., & Lin, Y. T. (2022). Deep Learning Anomaly Classification Using Multi-Attention Residual Blocks for Industrial Control Systems. Sensors, 22(23), 9084. https://doi.org/10.3390/s22239084

[20] Gawehn, P., Ergenc, D., & Fischer, M. (2022). Deep Learning-based Multi-PLC Anomaly Detection in Industrial Control Systems. GLOBECOM 2022 - 2022 IEEE Global Communications Conference, 4878-4884. https://doi.org/10.1109/globecom48099.2022.10001315

[21] Telikani, A., Shen, J., Yang, J., & Wang, P. (2022). Industrial IoT Intrusion Detection via Evolutionary Cost-Sensitive Learning and Fog Computing. IEEE Internet of Things Journal, 9(22), 23260-23271. https://doi.org/10.1109/jiot.2022.3188224

[22] Shtayat, M. M., Hasan, M. K., Sulaiman, R., Islam, S., & Khan, A. U. R. (2023). An Explainable Ensemble Deep Learning Approach for Intrusion Detection in Industrial Internet of Things. IEEE Access, 11, 115047-115061. https://doi.org/10.1109/access.2023.3323573

[23] Soliman, S., Oudah, W., & Aljuhani, A. (2023). Deep learning-based intrusion detection approach for securing industrial Internet of Things. Alexandria Engineering Journal, 81, 371-383. https://doi.org/10.1016/j.aej.2023.09.023

[24] Alani, M. M., Damiani, E., & Ghosh, U. (2022). DeepIIoT: An Explainable Deep Learning Based Intrusion Detection System for Industrial IOT. 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW), 169-174. https://doi.org/10.1109/icdcsw56584.2022.00040

[25] Bahadoripour, S., MacDonald, E., & Karimipour, H. (2023). A Deep Multi-Modal Cyber-Attack Detection in Industrial Control Systems. 2023 IEEE International Conference on Industrial Technology (ICIT), 1-6. https://doi.org/10.1109/icit58465.2023.10143147

[26] Jin, D., Chen, S., He, H., Jiang, X., Cheng, S., & Yang, J. (2023). Federated Incremental Learning based Evolvable Intrusion Detection System for Zero-Day Attacks. IEEE Network, 37(1), 125-132. https://doi.org/10.1109/mnet.018.2200349

[27] Bar, R., & Hajaj, C. (2022). SimCSE for Encrypted Traffic Detection and Zero-Day Attack Detection. IEEE Access, 10, 56952-56960. https://doi.org/10.1109/access.2022.3177272