International Journal of Applied Mathematics

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

FROM DETECTION TO DIAGNOSIS: TCAE–DBSCAN WITH LIME FOR INTERPRETABLE ICS ANOMALY ANALYSIS

Sangeeta Oswal[1], Subhash Shinde[2], Vijayalaksmi M[3]

[1,2]Department of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India

[3]Department of Artificial Intelligence and Data Science, Vivekanand Education Society Institute of Technology, Mumbai, India

Abstract

Industrial Control Systems (ICS) are frequent targets of cyberattacks, often leading to severe operational and safety consequences. Their continuous, automated operation with minimal human oversight increases the risk of undetected adversarial interference. While deep learning–based anomaly detection methods have shown strong performance in identifying abnormal behaviour, most lack the ability to pinpoint which sensors or actuators drive these detections. In this work, we propose a Temporal Convolutional Autoencoder (TCAE) combined with DBSCAN clustering to detect anomalies in the Secure Water Treatment (SWaT) dataset. The TCAE effectively captures long-term temporal dependencies and scales efficiently to large datasets. To enhance interpretability, we integrate Local Interpretable Model-Agnostic Explanations (LIME), which not only identifies the features that positively contribute to an anomaly but also highlights those with negative or negligible influence. This dual perspective enables domain experts to distinguish between primary drivers of abnormal behaviour and unaffected components, improving analysis, targeted mitigation, and user trust in automated detection systems.

Keywords: Industrial Control system, Temporal convolution network, LIME, SHAP, Explainable AI

1. Introduction

The rapid evolution of Industry 4.0 has led to the integration of cyber-physical systems, the Internet of Things (IoT), and cloud-based services into Industrial Control Systems (ICS) [1], [2]. These systems—such as water treatment plants, smart grids, and automated manufacturing lines—operate with minimal human intervention, making them highly efficient but also increasingly vulnerable to cyberattacks and operational faults. ICS environments typically comprise Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and Programmable Logic Controllers (PLCs), all of which must work reliably to ensure safe and continuous operation. Anomaly detection has emerged as a critical tool for safeguarding ICS against both malicious intrusions and unintentional failures. Deep learning–based approaches, particularly unsupervised models, have shown strong performance in identifying abnormal patterns in multivariate time-series data. However, despite their accuracy, these models often function as "black boxes," providing little insight into why a particular

Received: July 16, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

instance is flagged as anomalous. This lack of interpretability can hinder trust, slow incident response, and limit adoption by domain experts.

Explainable Artificial Intelligence (XAI)[3], [4] addresses this challenge by offering methods to interpret and visualize the decision-making process of complex models. Among the various XAI techniques, Local Interpretable Model-Agnostic Explanations (LIME) [5] has gained attention for its ability to approximate a model's local decision boundary and highlight the most influential features for individual predictions. Unlike global explanation methods, LIME focuses on instance-level interpretability, making it particularly suitable for analysis of specific anomalies in ICS environments.

In this study, we apply LIME to interpret anomalies detected in the Secure Water Treatment (SWaT)[6] testbed dataset a widely used benchmark for ICS security research. Our approach combines an unsupervised anomaly detection model with LIME-based explanations to identify not only when an anomaly occurs but also which sensors or actuators contribute most to it. By providing clear, instance-specific explanations, this work aims to enhance operator trust, support root-cause analysis, and improve the practical deployment of anomaly detection systems in critical infrastructure. This research employs the Secure Water Treatment (SWaT) testbed dataset to train and evaluate a Temporal Convolution Autoencoder (TCAE) model for anomaly detection. The TCAE leverages dilated temporal convolutions to effectively capture long-term dependencies in time-series data, enabling it to learn intricate temporal patterns. The model is trained exclusively on normal operational data, with the assumption that anomalous events will yield higher reconstruction errors. Detected anomalies are further refined using the DBSCAN clustering algorithm to isolate attack points. The novelty of this work lies in integrating LIME into the anomaly detection pipeline to provide transparent, instance-specific explanations for each detected attack. By identifying the most influential sensors and actuators contributing to an anomaly, the proposed method supports root-cause analysis and enhances operator trust. The contributions of this study are as follows:

- Application of LIME to interpret anomalies detected by a TCAE model in the SWaT dataset.
- Visualization of feature contributions for each detected attack, enabling targeted analysis.
- Demonstration of LIME's effectiveness in improving the interpretability and trustworthiness of anomaly detection in ICS.

2. Related Work

Anomaly detection approaches in industrial control systems (ICS) are generally grouped into two broad categories: distribution-based methods, which learn the statistical profile of normal operational data, and reconstruction-based methods, which flag instances with high reconstruction error as anomalous. In recent years, a variety of deep learning architectures have been applied to benchmark datasets such as SWaT, including autoencoders [7], generative adversarial networks (GANs)[8], [9], transformers [10], [11], and other sequence-learning models. These methods predominantly aim to improve detection accuracy, often with limited emphasis on explainability.

Given the high-dimensional, temporally correlated nature of ICS telemetry, deep learning models are frequently preferred over traditional machine learning techniques. Unsupervised

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

detection remains the dominant paradigm, under the assumption that normal behaviour is far more prevalent than anomalous events in operational data. Sequence-based models are particularly effective at capturing temporal dependencies, while more recent work extends this to spatial dependencies through transformer architectures and graph attention networks. Correlation-driven pipeline combines LSTM-autoencoder [12] windowing with latent correlation features and a multivariate Gaussian detector, on SWaT, HIL-HAI, and IoT Modbus datasets, while integrating SHAP for both feature selection and root-cause analysis. Other studies have explored hybrid deep-learning-plus-ensemble frameworks e.g., autoencoder feature selection with random forest classification[13] and BERT-based sequence models for ICS logs [14], reporting strong generalization and low false-alarm rates.

Despite these advances, a persistent gap remains in model transparency. Many existing works provide only global rankings of the top-k contributing features across all anomalies, offering limited insight into specific attack scenarios [15]. In contrast, our approach conducts attack-level attribution analysis, identifying the most influential features for each distinct incident, thereby enhancing interpretability and operational trust.

Detection Model	Methodology Used	Explainability Technique	Role of XAI in Analysis	Year
GAN-AD[16]	Generative Adversarial Networks	No		2018
DAEMON[15]	Adversarial Autoencoder Anomaly Detection Interpretation	Yes	The top-k dimensions with the largest reconstruction error will be returned as the root cause of the anomaly.	2021
FID-GAN[17]	fog-based, unsupervised intrusion detection system using GANs	No		
TranAD[10]	Transformers	No		
WaXAI[18]	(ECOD and DeepSVDD)	Yes	Compute the feature scores for SHAP, LIME, ALE, and IG, utilizing the anomaly scores	2024
CCTAK[19]	VAE with TCN	Yes	Proposed new	2024
LSTM-AE[12]	Correlation features: Gaussian distribution model	SHAP	evaluation matrix Analysis and root cause	2025

3. Methods

3.1 **Explainable AI**: Deep learning–based anomaly detection models in Industrial Control Systems (ICS) often operate as black boxes [20], producing accurate predictions without revealing the reasoning behind them. This opacity can hinder trust, slow incident response, and limit adoption in safety-critical domains. Explainable Artificial Intelligence (XAI) addresses this challenge by providing techniques to interpret and visualize model decisions, enabling

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

domain experts to understand why a model flagged a particular instance as anomalous. XAI methods can be broadly classified into:

- Model-specific: Designed for a particular class of models (e.g., decision trees, linear models).
- Model-agnostic: Applicable to any predictive model, regardless of its internal structure.

Model-agnostic methods, such as LIME and SHAP[21], are particularly valuable in ICS anomaly detection because they can be applied to complex architectures like autoencoders, transformers, or graph-based models without modifying the underlying training process.

LIME : (Local Interpretable Model-Agnostic Explanations)[5] explains individual predictions by approximating the original complex model with a simpler, interpretable surrogate model in the local neighbourhood of the instance being explained. The key idea is to perturb the input data around the instance of interest, observe the changes in the model's output, and fit a weighted interpretable model (e.g., linear regression) to these perturbed samples. Let:

- $f:R^d \rightarrow R$ the original anomaly detection model.
- $x \in R^d$ be the instance to explain.
- $\pi_x(z)$ be a proximity measure between z and x
- $g \in G$ be a candidate surrogate model from an interpretable family G.
- $\Omega(g)$ be a complexity penalty to maintain interpretability.
- LIME solves the following optimization problem:

$$L(f, g, \pi_x) + \Omega(g) \qquad (1)$$

where $L(f, g, \pi_x)$ measures the fidelity of g in approximating f in the locality defined by π_x, and $\Omega(g)$ encourages interpretability by penalizing overly complex surrogate models.

3.2 Model

The Secure Water Treatment (SWaT) dataset used in this study has a normal operation set (training) and an attack set (testing). The attack set contains both normal and abnormal records. The normal set is used to train the Temporal Convolutional Autoencoder (TCAE), while the attack set is used for evaluation. The underlying principle is that a model trained solely on normal behaviour will produce higher reconstruction errors when processing anomalous patterns. After generating predictions, we further analysed the results using explainability techniques to highlight the process variables most responsible for each detected anomaly. The following subsections describe the data preprocessing, anomaly detection model configuration, and explanation framework.

Data Preprocessing : The analysis was conducted on the SWaT dataset after removing all labels to enable unsupervised learning. All feature columns were converted to floating-point values and scaled to the range [0,1] using a min–max normalisation. To capture short-term temporal dependencies, a sliding window of length 12 was applied to the multivariate time series, producing sequences:

$$W=\{W_1,W_2,\ldots,W_t\} \qquad (2)$$

where each W_i is a matrix of shape (12,51), with 12 representing the time steps in the window and 51 the number of process variables. The resulting training set contained 494,988 windows, while the test set contained 449,907 windows.

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

TCAE Model: The proposed Temporal Convolutional Autoencoder (TCAE) is designed to learn the normal operational patterns of the plant. It uses dilated causal convolutional layers to capture dependencies over multiple time scales, enabling the model to detect deviations from expected behaviour. The architecture consists of:

- Encoder: Compresses the input sequence into a lower-dimensional latent representation.
- Decoder: Reconstructs the original sequence from the latent representation.

The model is trained to minimise reconstruction error. During testing, anomaly scores are computed using Kernel Density Estimation on the loss distribution, and DBSCAN is applied to identify low-density outliers as potential anomalies elaborated in section 3.3.

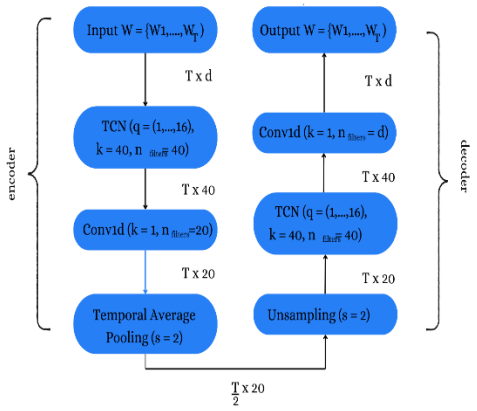

Figure 1. The TCAE Model

Figure 1 depicts the TCAE model. The encoder begins by examining the input windows. The TCN block consists of dilatation (q = 1,…,16) increased by a factor of 2. A one-dimensional convolutional layer [22] is subsequently employed to efficiently diminish the dimensionality of the feature map (TCN output). The series is reduced in resolution using an average pooling layer. The decoder executes the inverse function of the encoder to regenerate the input sequence, and the reconstruction loss is calculated.

3.3 Anomaly Detection

The detection process followed a multi-stage pipeline that combined reconstruction-based scoring with density-based clustering. First, the reconstruction loss for each time step was computed from the TCAE outputs. To better understand the statistical distribution of these loss values, Kernel Density Estimation (KDE) was applied using a Gaussian kernel. This step produced a smooth probability density function, highlighting regions of high and low data concentration. Next, the DBSCAN [23]algorithm was used to cluster the loss values based on their density. Points located in sparse regions of the distribution (cluster label = −1) are treated as potential anomalies. This approach reduced the likelihood of false positives from transient noise or minor fluctuations, as DBSCAN inherently filters out isolated points that do not belong to any dense cluster.

The resulting anomaly labels were visualised, with normal points shown in blue and anomalies in red (Figure 2). Each detected anomaly was then mapped to its corresponding timestamp and cross-referenced with the known attack periods in the dataset, as shown in Figure 3. This

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

mapping enabled a direct evaluation of detection performance by comparing predicted anomalies with ground-truth attack intervals. By pinpointing the start and end indices of each detected attack, the method supports a deeper investigation into the characteristics of anomalous segments.

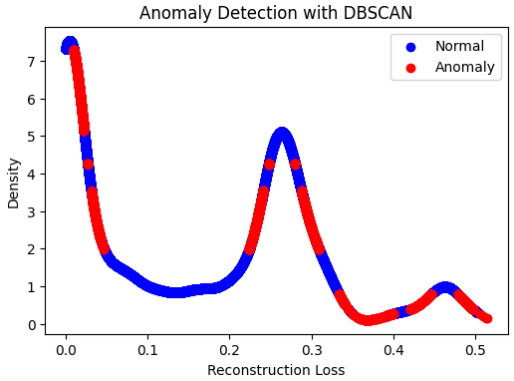

Figure 2: Density-based clustering of reconstruction errors using DBSCAN.

```
Attack Point: P-203, P-205
Start index: 17226, End index: 17258
---
Attack Point: LIT-401, P-401
Start index: 17291, End index: 17352
---
Attack Point: P-101, LIT-301
Start index: 19829, End index: 19974
---
Attack Point: P-302, LIT-401
Start index: 22782, End index: 22951
---
Attack Point: P-201, P-203, P-205
Start index: 27912, End index: 27924
```

Figure 3: Mapping of detected anomalies to attack points and time indices..

3.4 XAI using LIME

To identify the root cause of anomaly for identified attack all the window segments are considered and following step is performed

a. **Background Data Preparation** Use normal-operation windows$x_{norm} \in R^{N*T*F}$ flattened to$R^{N*(T.F)}$ as the explainer's training set.

b. **Predictor Wrapping** Define $Predict_{fn(z)} = \{l(z_i)\}_{i=1}^{|z|}$, where l reshape each z_i back to R^{T*F} and return reconstruction loss.

c. **Explainer Initialization** Instantiate LimeTabularExplainer in regression mode over the flattened feature space, with feature label

d. **Local Explanations** For each window index $i \in A_K$, generate an explanation vector $w_i \in R^{T.F}$ via $w_i = Lime.explain\_instance(x_i, predict_{fn})$

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Algorithm 1: LIME attribution for XAI

Initialization : $Attack\ Window\ Ak, TCAE\ Model\ l, background\ x_{norm}$

Output: $Attributio\ tensor\ wk\ \in R \wedge k\{L_k * T * F\}$

1. Flatten $x_{norm} \rightarrow\ z_{norm} \in R \wedge \{N * (T.F)\}$
2. explainer ← LimeTabularExplainer(z_{norm}, mode='regression', feature_names)
3. For i = a_k to b_k do
4. $x\_i\_flat\ \leftarrow reshape(window_i, 1, T.F)$
5. $explain_i \leftarrow explainer.explain\_instance(x_{i_{flat[0]}}, predict_{fn})$
6. $w\_i \leftarrow zeros(T.F)$
7. For (f_idx, weight) in exp_i.local_exp[output_index] do
8. $W\_i[f_{idx}] \leftarrow weight$
9. END
10. END

Return w_k

Once w_k are obtained for each segment, within-segment averaging is performed using

$$w_{t,f}^{k} = \frac{1}{L_k} \sum_{i}^{L_k} w_{i,t,f}^{k} \qquad (3)$$

And collapse time to get the global feature only

$$s_f^k = \frac{1}{T} \sum_{1}^{T} w_{t,f}^{k} \qquad (4)$$

e. **Ranking and Visualization** Sort s_f^k highlight the top positive and negative contributors for each attack k. Comparative bar charts or heatmaps can juxtapose the signature patterns of different attack types.

Algorithm 1 explain the procedure followed for LIME, in which for each detected attack segment A_k, the goal is to compute an attribution tensor $W^k \in R^{L_K * T * F}$ where L_k is the number of windows in the segment, T is the temporal length of each window, and F is the number of features. The procedure begins by flattening the background (normal) dataset x_{norm} of shape (N,T,F)into a two-dimensional array $z_{norm} \in R \wedge \{N * (T.F)$. This flattened representation is used to initialise a LimeTabularExplainer in regression mode, with feature names corresponding to each time–feature pair.

For every window index i in the attack segment, the corresponding time-series window is flattened into a vector of length T·F and passed to the explainer along with a prediction function that returns the model's reconstruction loss for reshaped inputs. LIME perturbs the input locally, queries the model, and fits a sparse linear surrogate to approximate the model's behaviour in the vicinity of that instance. The explainer returns a set of feature–weight pairs, where each weight quantifies the contribution of a specific time–feature combination to the anomaly score. These weights are stored in a zero-initialised vector of length T·F and then reshaped back to (T,F) to preserve temporal structure. Repeating this process for all L_k windows yields the full attribution tensor W^k for the segment.

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This tensor can subsequently be averaged across the temporal and/or window dimensions to produce aggregated feature importance scores for the entire attack, enabling comparison across different attack scenarios.

4. Results and Discussion

4.1 SWaT Dataset:

The research work was conducted on SWaT, a water treatment plant by iTrust Singapore to support research in cyber-physical system [6]

Table 2. The six stage sensors and actuators of SwaT dataset

Process	Sensor	Actuator
P1	LIT-101, FIT-101	MV-101, P101
P2	AIT-201, AIT-202, AIT-203, FIT-201	MV-201, P-201, P-202, P-203, P-204, P-205, P-206
P3	DPIT-301, FIT-301, LIT-301	MV-301, MV-302, MV-303, MV-304, P-301, P-302
P4	AIT-401, AIT-402, FIT-401, LIT-401	P-401, P-402, P-403, P-404, UV-401
P5	AIT-501, AIT-502, AIT-503, AIT-504, FIT-501, FIT-502, FIT-503, FIT-504, PIT-501, PIT-502, PIT-503	P-501, P-502
P6	FIT-601	P-601, P-602, P-603

A sequence of assaults was executed on SWaT to disrupt its standard functioning. The assaults conducted on the SWaT datasets are delineated in Table 2 and categorized as single point (SP) and multi-point (MP). In an SP attack, the assailant alters a single state variable, whereas in an MP attack, many state variables are compromised, and the associated measurements are falsified. A total of 41 attacks were executed, of which our suggested model identified 31 attacks.

Table 3. The Type of attack on SWaT dataset

Type of attack	No. of attack
Single Stage Single Point Attacks	23
Single Stage Multi Point Attacks	6
Multi Stage Single Point Attacks	4
Multi Stage Multi Point Attacks	3

4.2 XAI Result

We present the details for Attack no. 24, which is on the points P203 and P205, and Attack no. 29, which is on P-201, P-203, P-205.

Attack 24 was detected by the proposed TCAE–DBSCAN framework and corresponded to abnormal behaviour in the actuator P-203. During this period, the operational state of P-203 deviated from its expected pattern, indicating a possible manipulation or malfunction. This anomaly had downstream effects on the process flow, with potential implications for water distribution and quality.

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

To investigate the factors contributing to this detection, Local Interpretable Model-Agnostic Explanations (LIME) was applied to the windows associated with the attack interval. For each anomalous window, LIME generated local feature weights by approximating the model's behaviour with a sparse linear surrogate in the vicinity of the instance. These weights were then aggregated across the entire attack segment to obtain mean contribution scores for each process variable. The analysis revealed that P-203 had the highest positive contribution to the anomaly score, confirming its central role in the detection. Other variables, such as FIT-401 and MV-302, showed smaller but non-negligible contributions, suggesting secondary process interactions during the attack. The distribution of feature contributions is presented in Figure 4 as a box plot, where the median, interquartile range, and outliers illustrate the variability of each feature's influence across the attack windows.

This interpretability step not only validates the model's focus on the correct process component but also provides actionable insights for operators. By identifying P-203 as the dominant driver of the anomaly, the analysis supports targeted investigation and preventive maintenance strategies to mitigate similar incidents in the future.

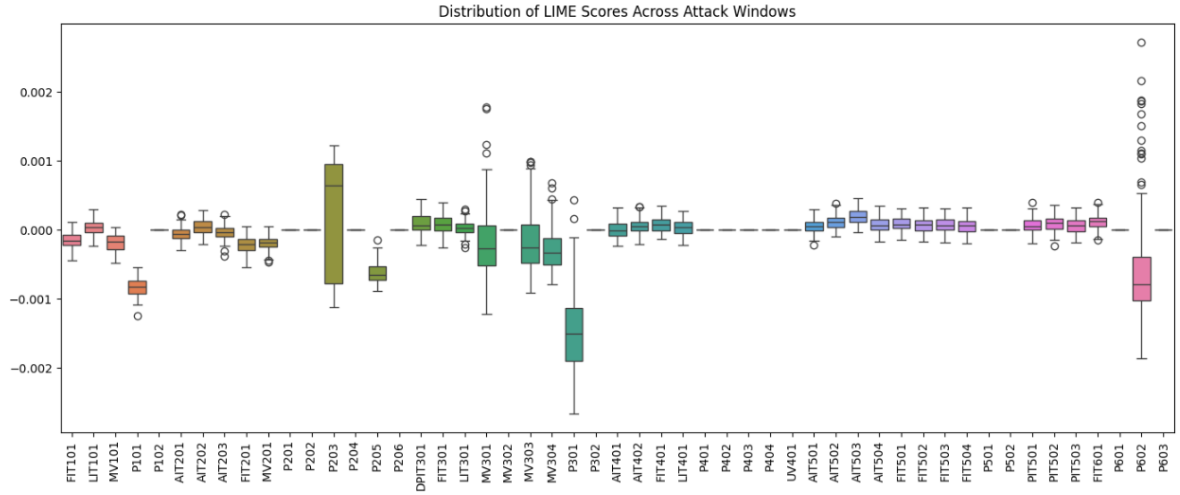

Figure 4. Box-Plot for LIME aggregation for Attack No 24

Attack 29: The aggregated LIME analysis highlights P-205 as the dominant positive contributor within this interval, Smaller but non-negligible effects appear on related flow/level measurements consistent with pump actuation, suggesting secondary process interactions during the attack. This pattern aligns with the attack specification and validates that the detector's decision is driven primarily by the manipulated actuators. Figure 5 presents the LIME contribution profile for the segment as a horizontal bar chart, separating features with negative (red) and positive (blue) influence. Bars correspond to per-feature contributions averaged across all windows in the attack interval; longer bars denote higher impact.

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

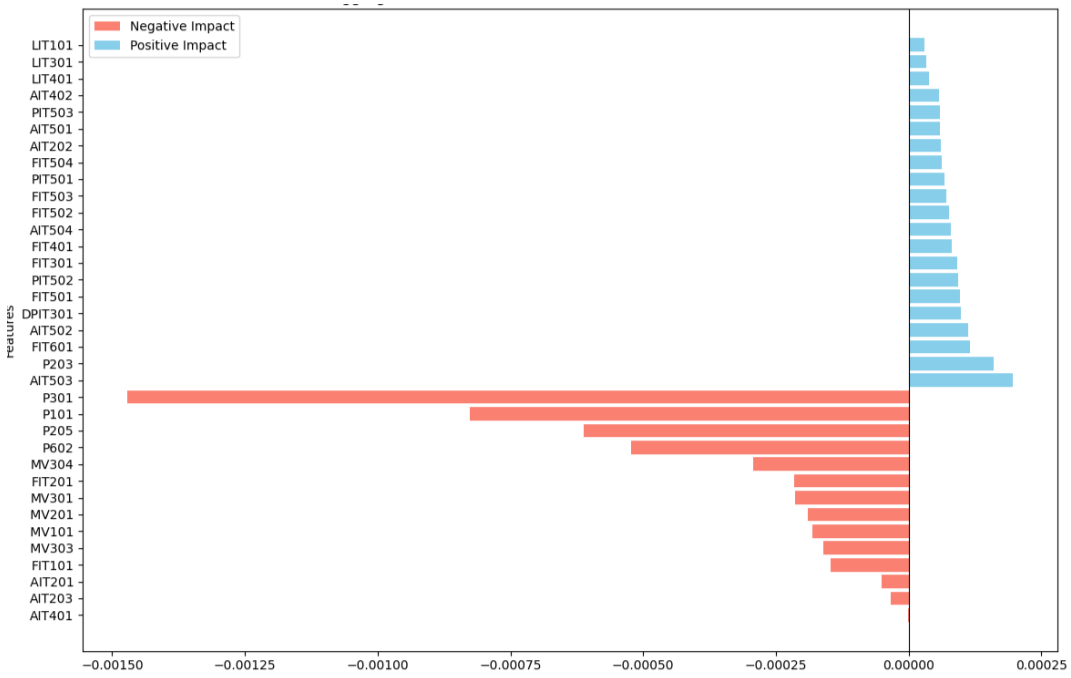

Figure 5. Aggregated LIME importance for Features for Attack No.29

5. Conclusion:

This study presented a Temporal Convolutional Autoencoder (TCAE)–based framework for detecting anomalies in industrial control systems, evaluated on the Secure Water Treatment (SWaT) dataset. By training exclusively on normal operational data, the model effectively identified deviations indicative of cyberattacks or process faults. The integration of DBSCAN clustering with reconstruction-loss analysis enhanced robustness by filtering out noise and isolating true anomalies.

A key contribution of this work is the incorporation of Local Interpretable Model-Agnostic Explanations (LIME) to provide transparent, instance-level insights into the model's decisions. For each detected attack segment, LIME identified the most influential process variables, enabling domain experts to trace anomalies back to specific sensors or actuators. This capability not only improves trust in automated detection systems but also supports targeted analysis and timely mitigation strategies. Experimental results demonstrated that the proposed approach successfully detected a substantial proportion of the documented attacks in the SWaT dataset, while offering interpretable explanations for each event.

Reference

[1] Y. Luo, Y. Xiao, V. Tech, G. Peng, and D. Yao, "Deep Learning-Based Anomaly Detection in Cyber-Physical Systems: Progress and Opportunities," 2021, doi: https://doi.org/10.1145/3453155.

[2] J. Giraldo et al., "A survey of physics-based attack detection in cyber-physical systems," ACM Comput. Surv., vol. 51, no. 4, p. 76, Jul. 2018, doi: 10.1145/3203245.

[3] A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti, "Explainable AI for Time Series Classification: A Review, Taxonomy and Research Directions," IEEE Access, vol. 10,

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

pp. 100700–100724, 2022, doi: 10.1109/ACCESS.2022.3207765.

[4] I. Šimić, V. Sabol, and E. Veas, "XAI Methods for Neural Time Series Classification: A Brief Review," Aug. 2021, doi: 10.48550/arxiv.2108.08009.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," NAACL-HLT 2016 - 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Demonstr. Sess., pp. 97–101, Feb. 2016, doi: 10.18653/v1/n16-3020.

[6] M. R. Gauthama Raman, W. Dong, and A. Mathur, "Deep autoencoders as anomaly detectors: Method and case study in a distributed water treatment plant," Comput. Secur., vol. 99, p. 102055, 2020, doi: 10.1016/j.cose.2020.102055.

[7] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: UnSupervised Anomaly Detection on Multivariate Time Series," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2020, pp. 3395–3404, doi: 10.1145/3394486.3403392.

[8] M. A. Bashar and R. Nayak, "TAnoGAN: Time Series Anomaly Detection with Generative Adversarial Networks," 2020 IEEE Symp. Ser. Comput. Intell. SSCI 2020, pp. 1778–1785, Aug. 2020, doi: 10.1109/SSCI47803.2020.9308512.

[9] T. Truong-Huu et al., "An Empirical Study on Unsupervised Network Anomaly Detection using Generative Adversarial Networks," in Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence, Oct. 2020, pp. 20–29, doi: 10.1145/3385003.3410924.

[10] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data," Jan. 2022, doi: 10.48550/arxiv.2201.07284.

[11] L. Yu, "DTAAD: Dual Tcn-Attention Networks for Anomaly Detection in Multivariate Time Series Data," 2023.

[12] E. Birihanu and I. Lendák, "Explainable correlation-based anomaly detection for Industrial Control Systems," Front. Artif. Intell., vol. 7, p. 1508821, Feb. 2024, doi: 10.3389/FRAI.2024.1508821/BIBTEX.

[13] X. K. Li, W. Chen, Q. Zhang, and L. Wu, "Building Auto-Encoder Intrusion Detection System based on random forest feature selection," Comput. Secur., vol. 95, p. 101851, Aug. 2020, doi: 10.1016/J.COSE.2020.101851.

[14] P. V Thali and V. Pachghare, "LLM-Based Detection of Cyber Anomalies in Industrial Control Systems."

[15] X. Chen et al., "DAEMON: Unsupervised Anomaly Detection and Interpretation for Multivariate Time Series," in 2021 IEEE 37th International Conference on Data Engineering (ICDE), Apr. 2021, vol. 2021-April, pp. 2225–2230, doi: 10.1109/ICDE51399.2021.00228.

[16] D. Li, D. Chen, J. Goh, and S. Ng, "Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series," 2018, doi: https://doi.org/10.48550/arXiv.1809.04758.

[17] P. Freitas de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. Gondim Santos, D. Macedo,

Received: July 16, 2025

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

and C. Zanchettin, “Intrusion Detection for Cyber–Physical Systems Using Generative Adversarial Networks in Fog Environment,” IEEE Internet Things J., vol. 8, no. 8, pp. 6247–6256, Apr. 2021, doi: 10.1109/JIOT.2020.3024800.

[18] K. Mathuros, S. Venugopalan, and S. Adepu, “WaXAI: Explainable Anomaly Detection in Industrial Control Systems and Water Systems,” ACM CPSS 2024 - Proc. 10th ACM Cyber-Physical Syst. Secur. Work., pp. 3–15, 2024, doi: 10.1145/3626205.3659147.

[19] Y. Abudurexiti, G. Han, F. Zhang, and L. Liu, “An explainable unsupervised anomaly detection framework for Industrial Internet of Things,” Comput. Secur., vol. 148, p. 104130, Jan. 2025, doi: 10.1016/J.COSE.2024.104130.

[20] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” IEEE Access, vol. 6, pp. 52138–52160, Sep. 2018, doi: 10.1109/ACCESS.2018.2870052.

[21] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, “Towards a Rigorous Evaluation of XAI Methods on Time Series,” Sep. 2019.

[22] S. Bai, J. Zico Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.”

[23] M. Çelik, F. Dadaşer-Çelik, and A. Ş. Dokuz, “Anomaly detection in temperature data using DBSCAN algorithm,” undefined, pp. 91–95, 2011, doi: 10.1109/INISTA.2011.5946052.

Received: July 16, 2025