Unveiling anomalies in industrial control systems: a kernel SHAP-based approach with temporal convolution autoencoder

Sangeeta Oswal[1,2], Subhash Shinde[1], Vijayalaksmi Murli[2]
[1]Department of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India
[2]Department of Artificial Intelligence and Data Science, Vivekanand Education Society Institute of Technology, Mumbai, India

Article Info

Article history:

Received Apr 19, 2025
Revised Oct 9, 2025
Accepted Nov 4, 2025

Keywords:

Anomaly detection
Auto encoder
Explainable AI
Industrial control system
Temporal convolution network
Unsupervised learning

ABSTRACT

Industrial control systems (ICS) are often the target of cyber-attacks, leading to undesirable consequences. ICSs operate without human supervision, making them vulnerable to adversaries. In recent years, numerous deep learning-based solutions have demonstrated their efficiency in detecting anomalies in ICSs. However, there is a lack of ability to pinpoint the sensors and actuators that contributed to the anomaly. In this research work, we use kernel Shapley additive explanations (SHAP) to explain anomalies detected by a temporal convolution autoencoder (TCAE). The proposed TCAE model handles the long-term dependency effectively and is computationally effective on a large dataset. A comprehensive explanation is provided, focusing on the feature that contributed to the anomaly for each identified attack. The SHAP values are extracted for each identified attack and visually depict the feature that contributed to the anomaly for each attack, helping the expert to handle the attack and build user trust.

Corresponding Author:

Sangeeta Oswal
Department of Computer Engineering, Lokmanya Tilak College of Engineering, Mumbai University
Sector-4, Plot No. 17 and 18, Vikas Nagar, Koparkhairane, Navi Mumbai, India
Email: sangeeta.oswal14@gmail.com

1. INTRODUCTION

Industry 4.0 [1] incorporates cyber-physical systems, IoT, and cloud computing into industrial control system (ICS) for enhanced autonomy and dependability. A typical example of an ICS is a smart power grid, a water treatment plant, or a water distribution system. It is primarily an automated system that integrates various types of control systems, including supervisory control and data acquisition systems (SCADA), distributed control systems (DCS), and programmable logic controllers (PLCs). ICSs operate in a smart environment and are subject to external threats. Anomaly detection using deep learning techniques [2], [3] is widely used in ICS for fault detection and handling failures of sensors and actuators. Numerous anomaly detection techniques have been created to identify cyberattacks in ICS. Although prior research on ICS domains has primarily focused on improving model performance, anomaly detection models have been challenging to analyze. This deficiency can hinder efforts to persuade experts to trust and implement potentially advantageous anomaly detection systems. The results of the anomaly detection algorithms may reveal anomalous cases previously unknown to the domain expert, and explaining the anomaly enhances the expert's belief in the algorithm.

A lack of explanation hinders decision-makers and domain specialists from using the algorithm output. A viable approach to address this problem is to employ explainable artificial intelligence (XAI) models [4] in conjunction with the anomaly detector. Thus, XAI [5], [6] is crucial for clarifying and understanding the black-box models. Numerous existing anomaly detection methods have integrated their models with XAI to improve transparency and reliability. Autoencoder-based frameworks, for end-to-end workflows for cooling-system anomaly detection with expert-guided localization [7], have demonstrated strong performance on real-world machinery data. In ICS network security, class-based SHAP analyses provide semi-global insights for network intrusion detection systems [8]. Log-centric approaches like the SXAD framework transform black-box log detectors into transparent white-box systems [9], and Siemens' industrial-grade XAI white paper outlines essential architectures and requirements for explainable manufacturing artificial intelligence (AI) [10]. However, prior research has not evaluated the dependability of XAI in the context of ICS security.

To deliver high-quality interpretation for anomaly localization at a low computational cost, we employ modern methods to identify anomalous data in the ICS domain. Additionally, we determine the effective XAI approach for explaining the anomalous data point and building user trust in deploying the model. This study uses the secure water treatment (SWaT) testbed dataset [11] to train and evaluate the anomaly detection model, temporal convolution autoencoder (TCAE), and assess its efficacy in localizing anomalies through XAI. We propose a TCAE, which employs a temporal convolution network as its foundation. TCAE excels in handling long-term dependency using dilation convolution to effectively learn long-term intricate temporal patterns. The density-based spatial clustering of applications with noise (DBSCAN) algorithm is used to flag the attack points, and we propose Kernel Shapley additive explanations (SHAP) [12], strategies to elucidate the models' decisions by calculating the SHAP values for each identified attack. The use of a lightweight TCAE architecture ensures scalability to real ICS environments, unlike heavier sequence models such as long short-term memory (LSTM) or Transformers normal data points. The TCAE model operates on the premise that anomalous data points exhibit elevated reconstruction loss, as the model is trained exclusively on normal data points. To our knowledge, no prior research has utilized SHAP to offer black-box explanations for abnormalities identified by an autoencoder on the SWaT dataset. The main contribution of this research work is: i) end-to-end fusion of a TCAE with Kernel SHAP for anomaly detection and explanation in ICS; ii) provides fine-grained explanations of the features that contribute most to anomalous behavior; iii) the use of a lightweight TCAE architecture ensures scalability to real ICS environments, unlike heavier sequence models such as LSTMs or Transformers; and iv) we evaluate the proposed approach on the SWaT dataset, a widely used ICS benchmark.

This research introduces a novel method utilizing SHAP values to elucidate the abnormalities detected in an autoencoder's output for each identified attack. Our pipeline flags anomalies at inception, pinpoints their root causes in raw measurement streams, and presents transparent, human-readable explanations that build operator trust and streamline decision-making. The TCAE+SHAP framework enables early anomaly detection by capturing subtle temporal deviations in ICS data, while SHAP explanations pinpoint root causes by attributing anomalies to specific sensors. Visual tools like force plots and heatmaps make these insights transparent, helping operators understand when, why, and where anomalies occur, building trust in AI-driven decisions. Together, these capabilities ensure early warning, rigorous root-cause analysis, and sustained trust, key prerequisites for resilient, mission-critical ICS operations. The method will be advantageous for specialists needing justification and visibility of anomalies. Domain specialists who utilized the explanations based on real-world data offered favorable comments, asserting that the explanations facilitated their comprehension and examination of the abnormalities.

The subsequent sections of this document are structured as follows: section 2 presents pertinent research on anomaly detection models and XAI within the ICS domain. The proposed model and explainable AI methodology employed in this study are discussed in section 3. Section 4 contains results, and finally, the conclusion is presented in section 5.

2. RELATED WORK

2.1. Anomaly detection

Anomaly detection algorithms can be separated into two types: distribution-based, which models the distribution of normal samples, and reconstruction-based, which considers data points with high reconstruction error to be anomalous. Several deep-learning methods [13] have been deployed to identify anomalies in the SWaT dataset. Methods utilizing autoencoders [14], generative adversarial network (GAN) [15], and Transformers [16] focus on enhancing model performance while placing less emphasis on XAI techniques. Due to ICS's intrinsic data, deep learning is preferred over machine learning models. Sequence learning models are employed to handle the temporal dependency of the data. Zhao *et al.* [17] proposed a gated recurrent unit (GRU)-based anomaly detector for SWaT, showing improved detection of attacks compared to CNN baselines. Kim *et al.* [18] used stacked LSTM layers with attention mechanisms to capture long-term dependencies in ICS telemetry. Li *et al.* [19] proposed a distangle time series to solve the Kullback-Leibler (KL) vanishing problem with LSTM. Zhang *et al.* [20] proposed a bidirectional LSTM with attention and built a convolution autoencoder model (CAE-M) to capture temporal dependency in the time series. Unsupervised Anomaly detection is favored over supervised techniques, which depend heavily on the assumption that normal cases are much more common than anomalous ones.

Spatial dependencies are better captured in models leveraging transformers and graph attention mechanisms. Tuli *et al.* [21] introduces a TranAD, transformer-based anomaly detection framework that leverages self-attention to model complex dependencies in ICS data such as SWaT and water distribution (WADI), outperforming recurrent architectures in scalability. Hybrid models like hybrid anomaly detection with multi-dimensional graph attention (HAD-MDGAT) [22] combine TCN with graph attention mechanisms to integrate temporal and relational dependencies across process variables. Recent surveys [2], [16] further highlight the growing adoption of Transformer and TCN variants in ICS security monitoring, reflecting a shift toward architectures that can scale with high-dimensional industrial time series. Researchers have proposed a model to reduce the training time for quick inference. Across all these models, the absence of a proper explanation and transparency to shed light on how and why a model has made a particular decision is a problem. Some of the research work only displays top k features contributing to all the anomaly points, whereas we have analyzed each attack and identified the features for each attack separately.

2.2. Explainable AI approach

XAI is a study domain concerning the transparency of an AI-based system [23]. Explainability refers to a system's ability to produce a set of features from an interpretable domain that impacted the decision for a particular instance. Conversely, certain research uses the terms explainability and interpretability interchangeably. This research focuses on the XAI technique deployed on the SWaT dataset. In this research paper, we used the term explainability to identify features that contributed to a reconstruction error. Table 1 provides a concise overview of recent research conducted.

XAI techniques such as SHAP, local interpretable model-agnostic explanations (LIME), integrated gradients (IG), and accumulated local effects (ALE) have been applied to time-series anomaly detection. In broader time-series contexts, Rojat *et al.* [6] demonstrated the application of SHAP for recurrent anomaly detection models, while Theissler *et al.* [4] surveyed interpretable deep anomaly detection approaches emphasizing feature-level attribution. Roshinta and Gabor [24] applied SHAP and LIME to multivariate sensor data, paving the way for their adoption in ICS domains. Bento *et al.* [25] introduced TimeSHAP, a variant of SHAP tailored for sequential data, enabling attribution of anomalies to specific time steps. Villani *et al.* [26] applied Kernel SHAP to LSTM outputs in ICS, showing how feature importance varies across attack phases. Fung *et al.* [27] evaluate multiple anomaly detection models on ICS datasets like SWaT and WADI, with emphasis on interpretability and reconstruction-based methods. Juttle *et al.* [28] proposed a C-SHAP-a concept-based SHAP extension for time series. It enables ICS operators to interpret model outputs using high-level temporal patterns and system topology overlays.

Our study contributes to this evolving research landscape by fusing a TCAE with Kernel SHAP to provide both accurate detection and fine-grained interpretability for ICS anomalies. Unlike prior works that primarily focus on detection accuracy, our approach highlights the specific sensors and control points most responsible for anomalous behavior during detected attack windows.

Table 1. The literature of anomaly detection using XAI

Model	Technique employed	XAI employed	Purpose of XAI	Year
USAD [14]	Autoencoders and GAN	No		2020
GAN-AD [29]	Generative adversarial networks	No		2018
DAEMON [30]	Adversarial autoencoder anomaly detection interpretation	Yes	The top-k dimensions exhibiting the highest reconstruction error will be identified as the primary source of the anomaly	2021
FID-GAN [31]	Fog-based, GANs	No		
HAD-MDGAT [22]	Graph attention network	No		
OCPAE [32]	One-class predictive autoencoder	No		2022
MAD_GAN [33]	GAN-LSTM/RNN	No		
TranAD [21]	transformers	No		
WaXAI [34]	(Deep SVDD and ECOD)	Yes	Derive LIME, ALE, SHAP, and IG feature scores	2024
CCTAK [35]	VAE with TCN and KAN	Yes	Proposed new evaluation matrix	2024

3. METHOD

3.1. Explainable AI

Due to the inherent capacity of black-box models [36], the demand for dependable explanations emerged. These explanations foster user trust, facilitate the identification of model failure sites, and eliminate obstacles to the implementation of deep neural networks across various fields. By developing more transparent and explicable systems, users will have a better understanding and, consequently, more trust in the model. Notable examples of techniques utilizing approximations include LIME [37], a model-agnostic approach for elucidating predictions via a local model, and DeepLIFT [38], a model-specific technique for interpreting deep learning models by backpropagating the contributions of all neurons to the input features. SHAP [12] integrates prior methodologies for elucidating predictions by quantifying feature significance, employing Shapley values from game theory to guarantee the consistency of the explanations. The SHAP framework proposes a model-agnostic method for approximating SHAP values, known as Kernel SHAP. Kernel SHAP employs linear LIME [37] in conjunction with Shapley values to construct a local explanation model. The local explanation model is a weighted linear regression constructed from a background dataset and a sample of potential feature coalitions in the data. SHAP specify explanation as in (1). In this context, (z') denotes the explanatory model, $\Phi j \in R$ represents the Shapley values for feature j, and each $z'j$ signifies a simplified value of the input features. $z' \in \{0, 1\}$ represents the coalition vector of the maximal coalition size M. In this context, $z'j$=1 indicates the presence of feature j inside the coalition, while its binary negation, $z'j$=0, signifies the lack of feature j. The Shapley value Φj can be computed in (2).

$$g(z') = \emptyset_0 + \sum_{j=1}^{m} \emptyset_j \, z'_j \tag{1}$$

$$\emptyset_j = \sum_{s\varepsilon \sqsubseteq F} \frac{|s|!(|F|-|s|-1)!}{|F|!} \left[f_{S\cup\{i\}} \left(x_{S\cup\{i\}}\right) - f_s(x_s)\right] \tag{2}$$

Let xs represent the values of the input features within the feature subsets set S, where all S are subsets of F, with F denoting the complete set of features. A model $fS\cup\{i\}$ is trained with the feature included, whereas a second model fs is trained without the feature. The expression $[fS\cup\{i\ \}(xS\cup\{i\ \})-fs(xs)]$ serves to compare the predictions of the two models. Since the impact of excluding a characteristic depends on other features, the aforementioned differences are computed for all viable subsets.

3.2. Model

The SWaT dataset comprises a normal (train) and an attack (test) dataset, with the latter containing both normal and attack points. The dataset is pre-processed, and the normal dataset is utilized for training the TCAE model. The attack dataset is used to generate predictions. The underlying premise is that a TCAE trained on a normal dataset will have higher reconstruction loss for anomalous data points. Subsequently, we evaluated the models' predictions and utilized XAI methods to clarify the outcomes related to the identified anomalies. The following sections outline the data preprocessing, configuration of the anomaly detection model, and the setup of XAI.

3.2.1. Data preprocessing

This study focused primarily on attacks on the SWaT datasets. Labels are eliminated for unsupervised processing, and the columns are transformed to floats and normalized with a min-max scaler. A 12-length local contextual window is used to convert the time series to a sliding window W={W1, W2, ..., Wt}. The entire training window size is 494988 (12, 51), while the test window size is 449907 (12, 51). In this case, 12 represents the window size, and 51 represents the dimension of the time series.

3.2.2. Temporal convolution autoencoder model

The proposed model TCAE shown in Figure 1 employs a TCN autoencoder to capture normal time stamps and uses this representation to identify abnormal patterns that deviate from expected behaviour. The model employs dilated convolutional layers and an expansive receptive field to examine the data across various temporal scales. The TCAE model facilitates the concurrent training of encoders and decoders. Encoders are designed to compress input time series, whereas decoders are responsible for reconstructing them. The reconstruction error serves as a tool for identifying anomalous behavior. The encoder comprises three temporal blocks, each using causal, dilated 1D convolutions with a doubling dilation schedule q∈ {1, 2, 4, 8, 16}, kernel size k=40, and 40 filters per layer, followed by residual connections for stability. Each convolution is followed by ReLU activation and weight normalization; channel compression is performed via a 1×1 Conv1d with 20 filters. Temporal down-sampling uses average pooling with a stride 2. The decoder mirrors the encoder: up-sampling (stride 2) restores temporal resolution, followed by dilated 1D convolutions (kernel size 40, 40 filters) and ReLU activation. A final 1×1 Conv1d projects back to 51 output channels with linear activation to reconstruct the input window. We optimize mean squared error (MSE) reconstruction loss over the full window (12×51); anomalies are scored as per-window MSE between input and reconstruction. The architectural details are provided in Table 2. The TCAE model is trained for 5 epochs using the Adam optimizer with a learning rate of 0.001 and held out 10 % of the normal training windows as a validation split; all convolutional kernels were initialized with Glorot normal. Table 3 summarizes the performance of the proposed model against existing approaches.

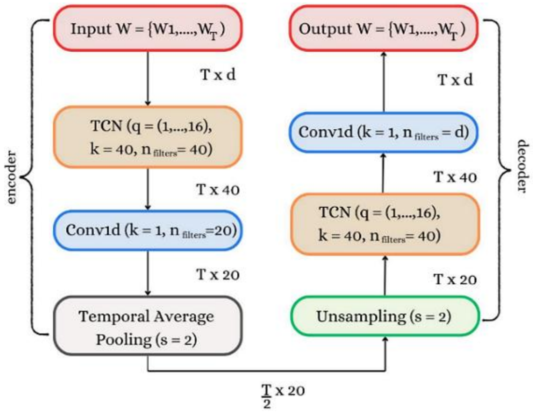

encoder
Input W = {W1,....,W_T}
T x d
TCN (q = (1,...,16), k = 40, n filters= 40)
T x 40
Conv1d (k = 1, n filters=20)
T x 20
Temporal Average Pooling (s = 2)
T/2 x 20
Unsampling (s = 2)
T x 20
TCN (q = (1,...,16), k = 40, n filters= 40)
T x 40
Conv1d (k = 1, n filters = d)
T x d
Output W = {W1,....,W_T}
decoder

Figure 1. The proposed TCAE model

Table 2. Summary of TCAE model architecture and hyperparameter settings

Component	Setting
Convolutional depth	3 encoder blocks+3 decoder blocks
Kernel sizes	40 (dilated conv), 1 (1×1 conv)
Dilations	1, 2, 4, 8, 16 (causal)
Filters	40 (temporal conv), 20 (1×1 compression), 51 output
Activations	ReLU (intermediate), Linear (output)
Pooling/up-sampling	Avg pool s=2/Nearest up sample s=2
Normalization	Weight normalization (temporal convs)
Residuals	Per-block identity skips
Loss	MSE over (12×51) window

Table 3. Model performance

Model	Precision	Recall	F1-score
DAGMM	0.4695	0.6659	0.5507
LSTM-NDT	0.7777	0.5108	0.6166
USAD	0.7488	0.5945	0.6627
LSTM-AE	0.8120	0.5780	0.6740
Isolation forest	0.6900	0.4600	0.5520
PCA	0.6200	0.4300	0.5050
TCAE (Ours)	0.9435	0.6136	0.7436

3.3. Anomaly detection

For identifying anomalous attack points, the reconstruction loss is identified on the test data. The model is trained exclusively on normal time series during the training phase. The TCAE model is based on the rationale that the model trained on normal data points will have low reconstruction loss, based on the understanding that the autoencoder should accurately reconstruct normal data points in the time series. When TCAE identifies patterns that deviate notably from the norm, we expect to see a rise in reconstruction loss. To identify anomalous data points, we employed a multi-step approach that combines reconstruction loss calculated using a TCAE and density-based clustering through DBSCAN.

To effectively identify anomalies, we leveraged a combination of kernel density estimation (KDE) and DBSCAN clustering. First, Gaussian_kde was applied to the reconstruction loss values, producing a density estimate that highlights the underlying distribution of the data. Subsequently, the DBSCAN algorithm was employed to group data points based on their density. This approach allowed for the identification of anomalies as points labeled as -1 by DBSCAN, corresponding to sparse regions in the data distribution. A visualization of the results depicted anomalies in red and non-anomalous points in blue, providing a clear distinction between clusters and outliers, as shown in Figure 2. This method not only identified outliers effectively but also facilitated the quantification of timestamps and attack points that are mapped to detected anomalies to the corresponding attack period. The attacks in the dataset were mapped to the anomalies flagged by the model. This mapping allowed us to measure detection performance by comparing the identified anomalies with the known attack instances. This methodology enables precise evaluation of anomaly detection models by providing insights into model performance in correctly identifying attack periods within the data. Figure 3 presents these outputs and highlights the precision with which the model identifies attacks within the dataset. By pinpointing the exact start and end indices of each attack, it becomes possible to analyze the characteristics of the detected anomalous segments and determine the underlying causes of the attacks. The ability to map detected anomalies back to specific attack points is crucial for validating the model's effectiveness and for providing actionable insights for system defense strategies.

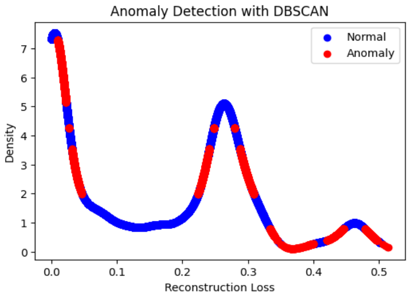
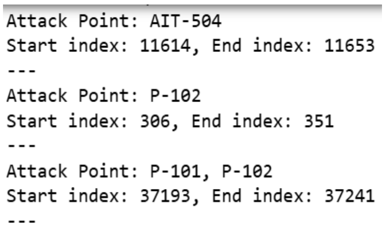


Figure 2. The result of DBSCAN on reconstruction loss



```
Attack Point: AIT-504
Start index: 11614, End index: 11653
---
Attack Point: P-102
Start index: 306, End index: 351
---
Attack Point: P-101, P-102
Start index: 37193, End index: 37241
---
```

Figure 3. A snapshot of detected attack points

3.4. Explainable AI using SHAP

While TCAE has been used for anomaly detection, its latent outputs are typically opaque to domain experts. To address this, we integrate Kernel SHAP post hoc to explain anomaly scores by attributing them to specific input features and time steps. This fusion enables temporal interpretability, allowing analysts to trace anomalies back to contributing sensors. Once the anomaly detection process identified attack windows and their respective indices, SHAP was employed to interpret the features contributing to each identified attack. To streamline computation, the background data was summarized using K-means clustering, reducing the dataset to 100 representative clusters (K=100). This clustering technique effectively captured the underlying feature space of the normal windows (windows_normal), ensuring computational efficiency and meaningful baseline comparisons.

Each attack window was flattened from its original multidimensional format (12-time steps×51 features) into a single-dimensional input for SHAP calculations. SHAP values were computed for selected attack windows (window indices) to quantify the feature importance for each anomalous instance. For interpretability, the SHAP values were reshaped back to their original dimensions (12×51), and the mean SHAP values across all analyzed windows were calculated to highlight features with significant contributions to the anomalies. The results were visualized using force plots and violin charts, providing a detailed ranking of features based on their mean SHAP values. This method pinpoints the most influential elements of each attack to explain observed anomalies and improve the anomaly detection model's interpretability. Such insights facilitate a deeper understanding of system vulnerabilities and pave the way for targeted mitigation. The complete workflow used is described in Figure 4. According to Figure 4, the process of anomaly identification and the use of XAI in the identified attack window is described in detail.

To adapt Kernel SHAP to sequential TCAE embeddings, each anomalous window (12×51) is first flattened into a 612-dimensional vector. We use the K=100 cluster centroids (from the normal windows) as background samples, and SHAP internally generates approximately 9,400 coalition samples per instance to fit the local linear explanation model. Each SHAP call requires one forward pass per coalition sample, so explaining a single window entails roughly 9,400 model evaluations. On our hardware (a single NVIDIA A100 GPU), computing SHAP values for one window takes approximately 40 seconds.

3.5. Deployment context

In its current form, the TCAE+kernel SHAP pipeline runs offline in a Python environment, but it is architected as a standalone inference service for ICS networks. In a production setting, the trained model and explainer are packaged (in a Docker container) and deployed on on-premise servers or edge gateways that already receive live sensor and actuator streams. The service continuously ingests timestamped measurements, computes per-window reconstruction losses to flag anomalies, and invokes Kernel SHAP to produce ranked, time-step–level attributions. Alerts containing the anomaly score and contributing variables can be emitted as JSON over REST APIs or message queues into existing monitoring dashboards or alarm systems. Security and compliance align with NIST SP 800-82 and IEC 62443 by enforcing encrypted telemetry channels, role-based access controls, and immutable audit logs for all detection and explanation events. Future work will validate live testbed integration, benchmark end-to-end latency and throughput, and quantify the impact of explainable alerts on operator decision cycles through controlled user studies.

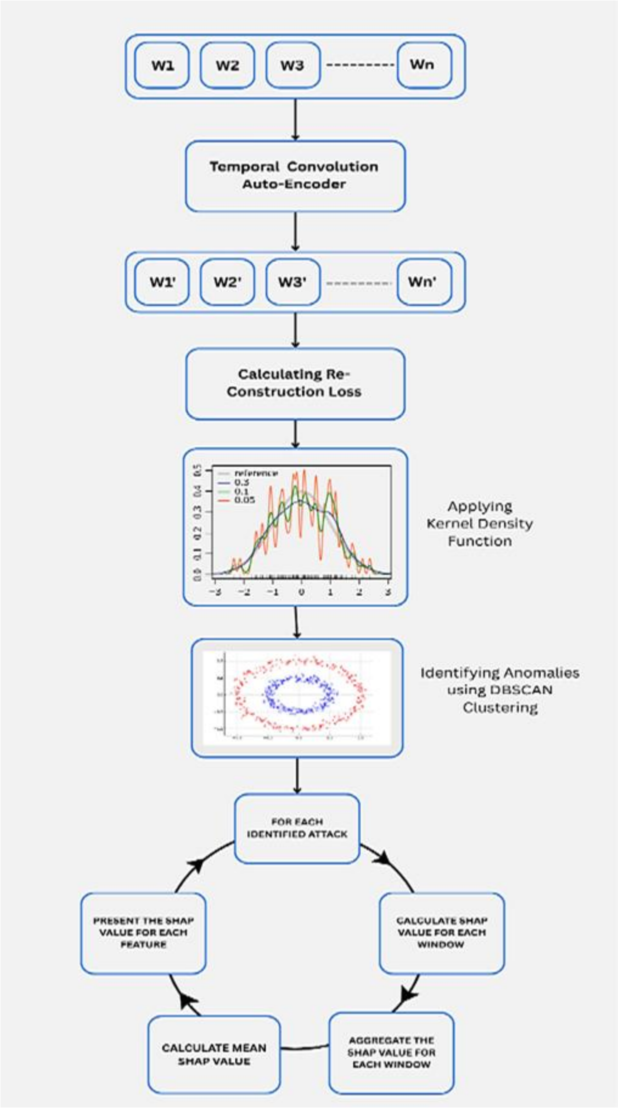

W1
W2
W3
Wn
Temporal Convolution Auto-Encoder
W1'
W2'
W3'
Wn'
Calculating Re-Construction Loss
reference
0.3
0.1
0.05
Applying Kernel Density Function
Identifying Anomalies using DBSCAN Clustering
FOR EACH IDENTIFIED ATTACK
CALCULATE SHAP VALUE FOR EACH WINDOW
AGGREGATE THE SHAP VALUE FOR EACH WINDOW
CALCULATE MEAN SHAP VALUE
PRESENT THE SHAP VALUE FOR EACH FEATURE

Figure 4. Flowchart of the proposed model and XAI method

4. RESULTS AND DISCUSSION

In this section, the results are presented. While the proposed method offers high-resolution attribution within identified attack windows, its interpretability is limited by the temporal granularity of SHAP explanations; sensor interactions outside the selected window may be underrepresented. Additionally, model drift poses a challenge in evolving ICS environments, where shifts in sensor behavior or operational patterns may degrade detection performance over time. We discuss in detail two attacks detected by our system and its SHAP values to identify the features contributing to the anomalies.

4.1. Secure water treatment dataset

The research work was conducted on SWaT, a water treatment plant developed by iTrust Singapore, to advance research in cyber-physical systems [39]. The SWaT dataset consists of 6 stages, P1 to P6, with various sensors and actuators as described in Table 4. It includes 51 synchronized variables (25 sensors, 26 actuators) covering flow, level, pressure, chemical analyzers, pumps, motorized valves, and UV units, recorded at a sampling rate of 1 Hz.

A series of attacks was launched on SWaT to disturb its normal operation. The attacks carried on the SWaT datasets are described in Table 5 and classified as single point (SP) and multi-point (MP). In an SP attack, the attacker manipulates one state variable, whereas in an MP attack, more than one state variables are compromised and the corresponding measurements are spoofed. The dataset contains 41 documented attack scenarios spanning single- and multi-stage as well as single- and multi-point manipulations; our proposed model correctly detected 31 of these attacks.

Table 4. The SWaT dataset sensors and actuators

Stages	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

Table 5. Attacks on SWaT dataset

Type of attack	Number of attacks
SSSP attacks	23
SSMP attacks	6
MSSP attacks	4
MSMP attacks	3

4.2. Explainable AI results

We present the XAI result on the attacks identified by our model, TCAE. In this research work, we elaborate on the feature contribution to each identified attack, unlike previous work, which presents only the top k feature contributions. We present in detail two attacks, Attack number 6, which is a single-stage single-point attack (SSSP) on the sensor AIT202, and attack number 22, which is a multi-stage multi-point attack (MSMP) on sensors and actuators in stages 4 and 5.

Attack 6 observed an anomaly in the parameter AIT-202, where its value exceeded 7.05. A corrective action was initiated by setting the value of AIT-202 to 6, yet no drainage process was triggered. This resulted in downstream effects such as the shutdown of P-203 and a subsequent change in water quality. To understand the features contributing to this anomaly, SHAP values were computed for the identified attack window. The calculated mean SHAP values revealed that AIT-202 had the highest contribution to the anomaly, with a mean SHAP value of 9.565626e-03, followed by P-203 (7.016715e-04), and other features such as P-602, FIT-601, and P-205, which exhibited significantly smaller contributions. This suggests that the anomalous behavior in AIT-202 was the primary driver of this attack scenario, while the influence of other features was negligible in comparison.

The SHAP analysis was instrumental in quantifying the feature importance for the detected anomaly. By ranking the mean SHAP values, it became evident that AIT-202's deviation was directly correlated with the observed impact on the system. Visualization of the feature contributions using SHAP force plots in Figure 5 and violin charts in Figure 6 provided further clarity on the significance of AIT-202 and its relationship to other features during the attack. This interpretability highlights the critical role of AIT-202 in the attack dynamics and underscores the need for enhanced monitoring of this parameter to mitigate similar incidents in the future. Figure 7 presents the SHAP value heatmap and sensor-wise attribution plot, visually reinforcing the dominant role of AIT-202 and highlighting the relative insignificance of other features during attack 6.

Attack 23 involves anomalies in the parameters UV-401, AIT-502, and P-501. The scenario was characterized by the following conditions: UV-401 was active, AIT-502 recorded a value below 150, and P-501 remained open. To interpret the feature contributions for this anomaly, SHAP values were computed for the identified attack window. SHAP summary plots and visualizations provided a clear ranking of feature contributions, offering insights into the dynamics of the anomaly. These findings underscore the necessity for closely monitoring UV-401 and P-501 during critical operations to prevent recurrence and mitigate potential risks. The force plot shown in Figure 8 presents the feature contribution to attack 23, and the violin plot in Figure 9 shows the distribution of SHAP values for each feature. The interpretability provided by SHAP enhances the understanding of system vulnerabilities and supports the development of targeted countermeasures for similar scenarios.

Figure 10 shows the SHAP heatmap for attack 23, highlighting the temporal importance of features like AIT-502, P-501, and UV-401. Figure 11 ranks sensors by their mean SHAP values, confirming these as the top contributors to the anomaly. Unlike existing approaches that merely present a ranked list of top-k contributing features without detailed context, our methodology provides granular insights into individual attack windows. This allows for targeted analysis and tailored mitigation strategies for each anomaly. By focusing on feature contributions specific to each attack, we ensure a comprehensive understanding of the root causes.

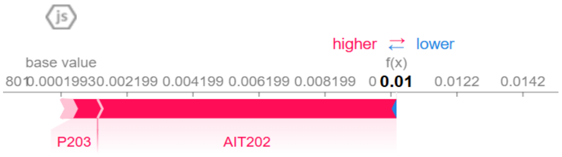


Figure 5. Force plot for attack number 6

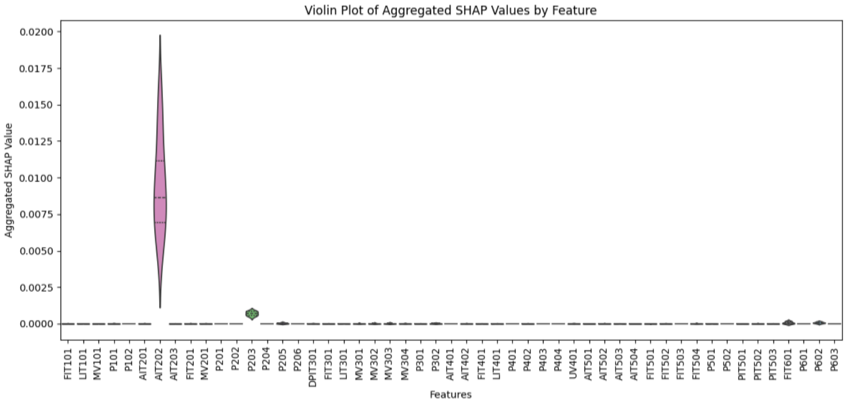


Figure 6. Violin Plot for feature contribution for attack number 6

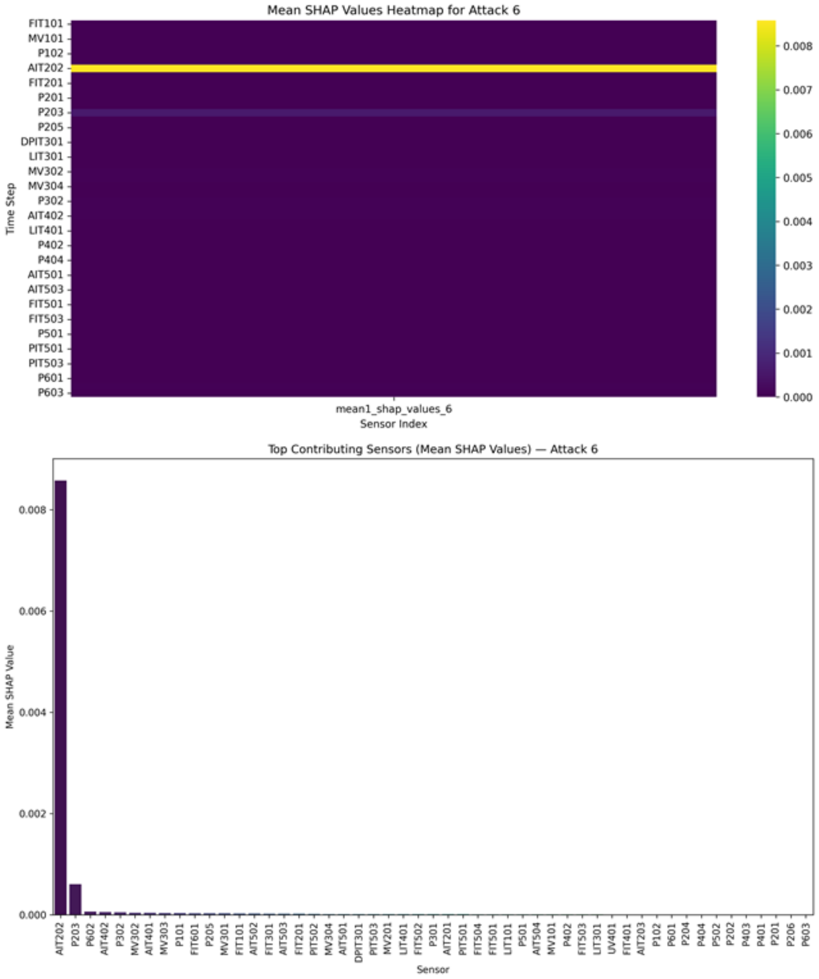


Figure 7. Heatmap and sensor-wise attribution for attack no 6

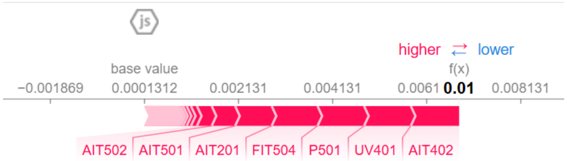


Figure 8. Force plot for attack number 23

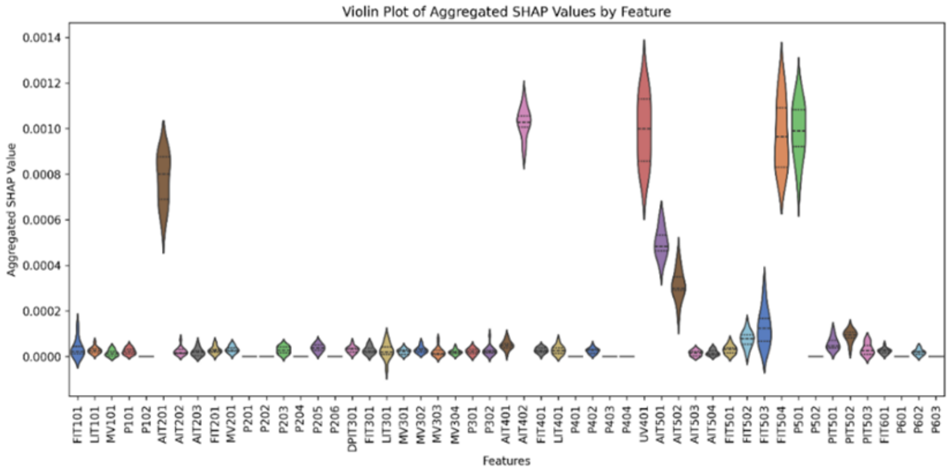


Figure 9. Violin Plot for SHAP values for attack number 23

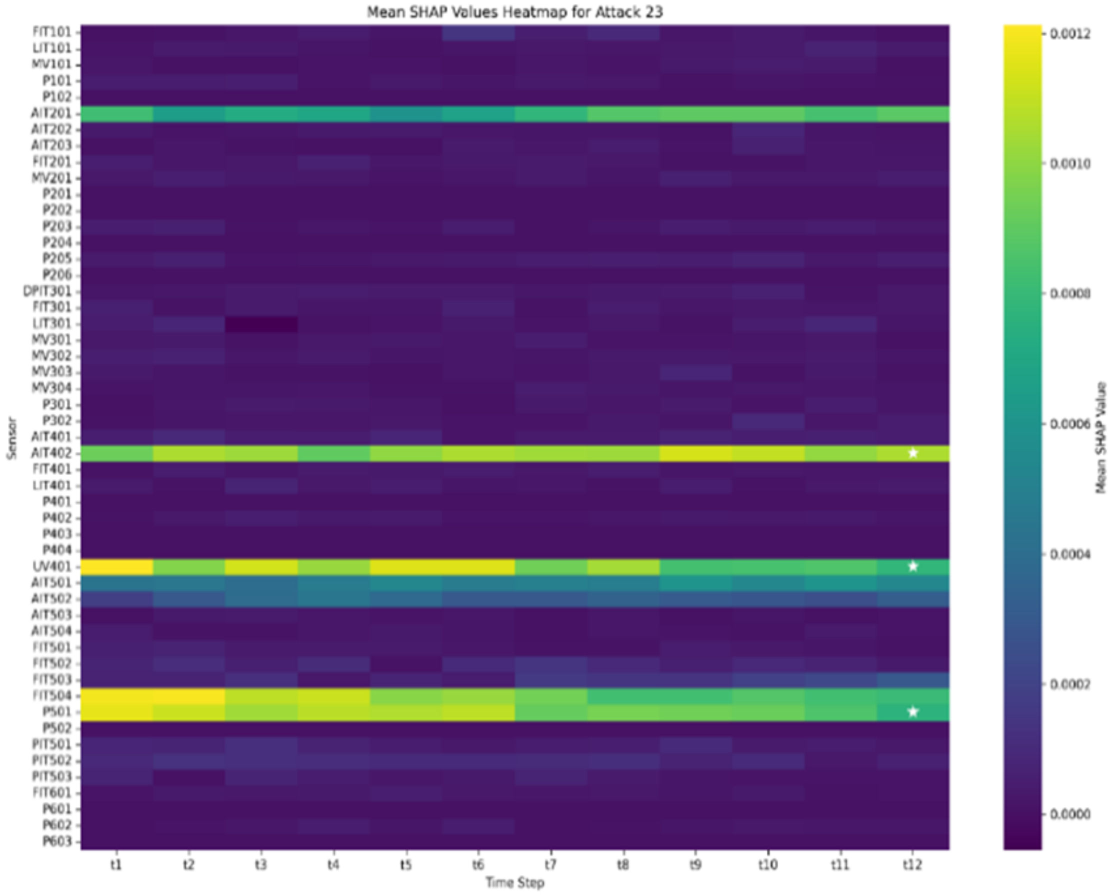


Figure 10. Heatmap for SHAP values attack 23

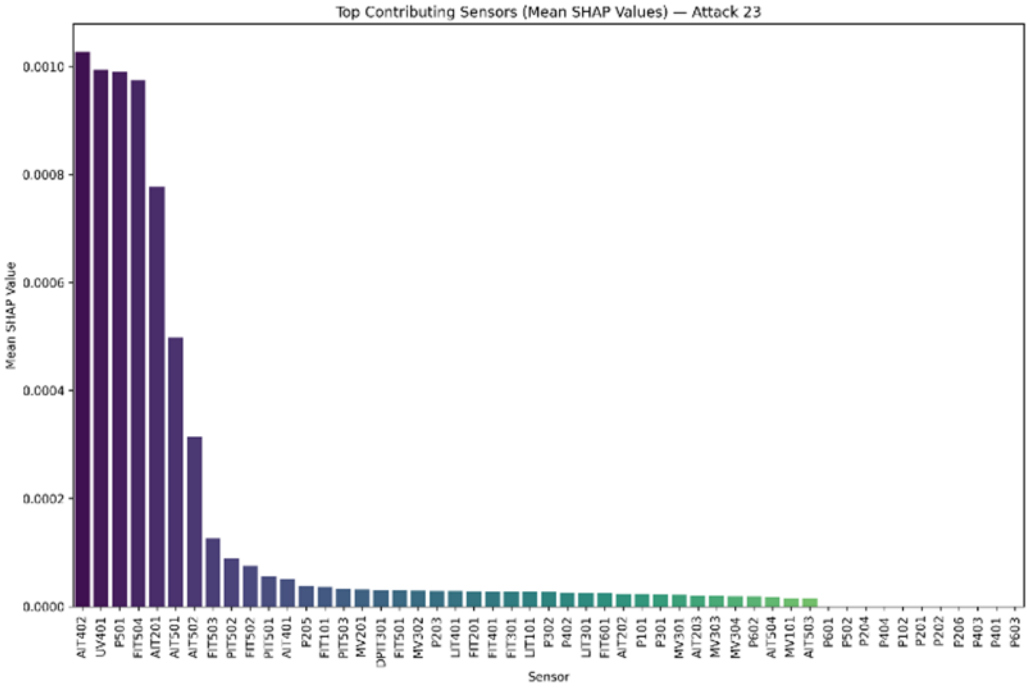


Figure 11. Sensorwise attribution for SHAP values attack 23

5. CONCLUSION

This study demonstrates the effectiveness of XAI techniques, particularly SHAP, in analyzing and interpreting anomalies detected in industrial systems. By leveraging SHAP, the black-box nature of advanced anomaly detection models is mitigated, providing transparent explanations for the contributions of individual features to detected anomalies. This clarity significantly enhances the trust and confidence of end users and stakeholders in the model's predictions and decisions. Furthermore, SHAP's interpretability empowers users to not only understand the root causes of detected anomalies but also make informed decisions to address and prevent similar incidents. The ability to identify attack scenarios and explain their underlying causes with precision highlights the potential of SHAP as a key tool in improving the robustness and reliability of anomaly detection systems. By bridging the gap between sophisticated algorithms and human interpretability, XAI ensures accountability and fosters confidence in deploying machine learning models for critical industrial applications. Future work will focus on four key extensions. First, we will incorporate online learning to adapt both the TCAE model and the SHAP explainer to gradual process drift without full retraining. Second, we plan to develop a privacy-preserving federated XAI framework to collaboratively refine anomaly models across multiple ICS sites. Third, integration with digital-twin platforms will enable synthetic fault generation, "what-if" analyses, and targeted operator training. Finally, we will validate the end-to-end pipeline on live ICS testbeds, measuring inference latency and throughput and conducting user studies to quantify the impact of explainable alerts on operator decision making.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sangeeta Oswal	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Subhash Shinde		✓				✓		✓	✓	✓	✓	✓	✓	
Vijayalaksmi Murli	✓		✓	✓		✓				✓	✓	✓	✓	

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding acquisition
Fo	: **Fo**rmal analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The SWaT dataset has to be requested from the iTrust Website at https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/.

REFERENCES

[1] Y. Zacchia Lun, A. D'Innocenzo, F. Smarra, I. Malavolta, and M. D. Di Benedetto, "State of the art of cyber-physical systems security: an automatic control perspective," *Journal of Systems and Software*, vol. 149, pp. 174–216, Mar. 2019, doi: 10.1016/j.jss.2018.12.006.

[2] Y. Luo, Y. Xiao, L. Cheng, G. Peng, and D. D. Yao, "Deep learning-based anomaly detection in cyber-physical systems: progress and opportunities," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–36, Jun. 2022, doi: 10.1145/3453155.

[3] J. Giraldo *et al.*, "A survey of physics-based attack detection in cyber-physical systems," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–36, Jul. 2019, doi: 10.1145/3203245.

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

[5] U. Schlegel, D. Oelke, D. A. Keim, and M. El-Assady, "An empirical study of explainable AI techniques on deep learning models for time series tasks," *arXiv*, Dec. 2020.

[6] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. D.-Rodríguez, "Explainable artificial intelligence (XAI) on timeseries data: a survey," *arXiv,* Apr. 2021.

[7] S. Holly *et al.*, "Autoencoder based anomaly detection and explained fault localization in industrial cooling systems," *PHM Society European Conference*, vol. 7, no. 1, pp. 200–210, Jun. 2022, doi: 10.36001/phme.2022.v7i1.3349.

[8] M. Pawlicki, A. Pawlicka, S. Szelest, R. Kozik, and M. Choraś, "Class-based SHAP analysis for improved explainability insights in NIDS," in *Communications in Computer and Information Science*, vol. 2387 CCIS, 2025, pp. 303–313. doi: 10.1007/978-981-96-1907-8_29.

[9] K. Alam, K. Kifayat, G. A. Sampedro, V. Karovic, and T. Naeem, "SXAD: Shapely explainable AI-based anomaly detection using log data," *IEEE Access*, vol. 12, pp. 95659–95672, 2024, doi: 10.1109/ACCESS.2024.3425472.

[10] M. Lebacher, R. Gross, and S. H. Weber, "The rise of industrial explainable artificial intelligence (XAI)–insights across the AI life cycle," *Siemens*, 2023.

[11] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10242 LNCS, pp. 88–99, 2017, doi: 10.1007/978-3-319-71368-7_8.

[12] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Computer Science > Artificial Intelligence*, May 2017, [Online]. Available: http://arxiv.org/abs/1705.07874

[13] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: a survey," *arXiv*, Jan. 2019.

[14] 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.

[15] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in time series: a survey and taxonomy," *arXiv,* 2021.

[16] Q. Wen *et al.*, "Transformers in time series: a survey," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2023-Augus, pp. 6778–6786, 2023, doi: 10.24963/ijcai.2023/759.

[17] P. Zhao, X. Chang, and M. Wang, "A novel multivariate time-series anomaly detection approach using an unsupervised deep neural network," *IEEE Access*, vol. 9, pp. 109025–109041, 2021, doi: 10.1109/ACCESS.2021.3101844.

[18] J. Kim, H. Kang, and P. Kang, "Time-series anomaly detection with stacked Transformer representations and 1D convolutional network," *Engineering Applications of Artificial Intelligence*, vol. 120, 2023, doi: 10.1016/j.engappai.2023.105964.

[19] Y. Li *et al.*, "Towards learning disentangled representations for time series," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3270–3278, 2022, doi: 10.1145/3534678.3539140.

[20] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 2118–2132, 2023, doi: 10.1109/TKDE.2021.3102110.

[21] S. Tuli, G. Casale, and N. R. Jennings, "TranAD," *Proceedings of the VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, May 2022, doi: 10.14778/3514061.3514067.

[22] L. Zhou, Q. Zeng, and B. Li, "Hybrid anomaly detection via multihead dynamic graph attention networks for multivariate time series," *IEEE Access*, vol. 10, pp. 40967–40978, 2022, doi: 10.1109/ACCESS.2022.3167640.

[23] A. B. Arrieta *et al.*, "Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[24] T. A. Roshinta and S. Gábor, "A comparative study of LIME and SHAP for enhancing trustworthiness and efficiency in explainable AI systems," in *2024 IEEE International Conference on Computing, ICOCO 2024*, Dec. 2024, pp. 134–139. doi: 10.1109/ICOCO62848.2024.10928183.

[25] J. Bento, P. Saleiro, A. F. Cruz, M. A. T. Figueiredo, and P. Bizarro, "TimeSHAP: explaining recurrent models through sequence perturbations," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2021, pp. 2565–2573. doi: 10.1145/3447548.3467166.

[26] M. Villani, J. Lockhart, and D. Magazzeni, "Feature importance for time series data: improving kernelSHAP," *arXiv*, 2022.

[27] C. Fung, S. Srinarasi, K. Lucas, H. B. Phee, and L. Bauer, “Perspectives from a comprehensive evaluation of reconstruction-based anomaly detection in industrial control systems,” in *European Symposium on Research in Computer Security*, 2022, pp. 493–513. doi: 10.1007/978-3-031-17143-7_24.

[28] A. Jutte, F. Ahmed, J. Linssen, and M. van Keulen, “C-SHAP for time series: an approach to high-level temporal explanations,” *arXiv,* 2025.

[29] D. Li, D. Chen, J. Goh, and S. Ng, “Anomaly detection with generative adversarial networks for multivariate time series,” *arXiv,* Jan. 2018.

[30] X. Chen *et al.*, “DAEMON: unsupervised anomaly detection and interpretation for multivariate time series,” in *Proceedings - International Conference on Data Engineering*, Apr. 2021, vol. 2021-April, pp. 2225–2230. doi: 10.1109/ICDE51399.2021.00228.

[31] P. F. D. Ar.-Filho, G. Kaddoum, D. R. Campelo, A. Gondim Santos, D. Macedo, and C. Zanchettin, “Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment,” *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6247–6256, Apr. 2021, doi: 10.1109/JIOT.2020.3024800.

[32] H. Zhang, F. Cheng, and A. Pandey, “One-class predictive autoencoder towards unsupervised anomaly detection on industrial time series,” in *KDD 2022 Workshop on Anomaly and Novelty Detection, Explanation, and Accommodation (ANDEA)*, 2022.

[33] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. K. Ng, “MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks,” in *International Conference on Artificial Neural Networks*, Jan. 2019, pp. 703–716. doi: 10.1007/978-3-030-30490-4_56.

[34] K. Mathuros, S. Venugopalan, and S. Adepu, “WaXAI: explainable anomaly detection in industrial control systems and water systems,” in *ACM CPSS 2024-Proceedings of the 10th ACM Cyber-Physical System Security Workshop*, Jul. 2024, pp. 3–15. doi: 10.1145/3626205.3659147.

[35] Y. Abudurexiti, G. Han, F. Zhang, and L. Liu, “An explainable unsupervised anomaly detection framework for industrial internet of things,” *Computers and Security*, vol. 148, p. 104130, Jan. 2025, doi: 10.1016/j.cose.2024.104130.

[36] A. Adadi and M. Berrada, “Peeking inside the black-box: a survey on explainable artificial intelligence (XAI),” *IEEE Access*, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[37] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?’ explaining the predictions of any classifier,” in *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 2016, pp. 97–101. doi: 10.18653/v1/n16-3020.

[38] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” *34th International Conference on Machine Learning, ICML 2017*, vol. 7, pp. 4844–4866, 2017.

[39] M. R. G. Raman, W. Dong, and A. Mathur, “Deep autoencoders as anomaly detectors: method and case study in a distributed water treatment plant,” *Computers & Security*, vol. 99, p. 102055, Dec. 2020, doi: 10.1016/j.cose.2020.102055.

BIOGRAPHIES OF AUTHORS

Sangeeta Oswal is pursuing a Ph.D. in Computer Science and Engineering from Mumbai University. She is currently an Assistant Professor at the Artificial Intelligence and Data Science Department in VESIT-Mumbai, India. She is also a NVIDIA DLI-certified instructor for 5 courses and is a gold-tier ambassador. Her research includes anomaly detection, deep learning, time series, LLM fine-tuning, and explainable AI. She has published over 30 papers in international journals and conferences. Her contribution to the research work includes writing the original draft, software, data curation, validation, and experimentation. She can be contacted at email: sangeeta.oswal@ves.ac.in.

Dr. Subhash Shinde is the principal of Lokanya Tilak College of Engineering, Navi Mumbai. He has published about 48 papers in international journals and conferences and has a copyright to his credit. He has also authored four books through reputed publishers. Under his supervision, 6 research scholars (Ph.D.) and 33 PG (ME) students have completed their degrees from the University of Mumbai. He has published about 60 papers in international journals and conferences. Currently, he is Chairman, Board of Studies in Computer Engineering under the Faculty of Science and Technology, as well as a member of the Research and Recognition Committee (RRC), University of Mumbai. His contribution to the current research work includes conceptualization, writing-reviewing and editing, supervision. He can be contacted at email: skshinde@ltce.in.

Dr. Vijayalaksmi Murli is the Vice Principal of Vivekanand Education Society’s Institute of Technology, Mumbai. She has completed her Ph.D. from IIT Bombay. She is a reviewer for several international journals and was a board of studies member for information technology at Mumbai University. Her research includes time series, LLM fine-tuning, and AI in healthcare. Her contribution to the research work includes investigation, writing, reviewing, and editing. She can be contacted at email: m.vijayalakshmi@ves.ac.in.