Interpretable Anomaly Detection in Cyber-Physical Systems via Sparse Autoencoder

1st Xin Wang
Computer Science Department
Vanderbilt University
Nashville, US
xin.wang.1@vanderbilt.edu

***Abstract*—This project proposes Sparse Autoencoders (SAEs) for interpretable anomaly detection in Cyber-Physical Systems (CPS). While current deep learning approaches demonstrate promising performance in detecting anomalies within complex CPS environments, they lack the essential interpretability needed to assist human operators in decision-making. I propose using SAEs to impose sparsity in hidden layers, creating disentangled representations corresponding to specific physical components or behaviors. This approach will be evaluated on benchmark CPS anomaly detection datasets, such as BATADAL, to demonstrate how sparse representations enhance detection accuracy and interoperability. The code is available at https://github.com/xwang112358/SAE-CPS.**

***Index Terms*—Anomaly Detection, Sparse Autoencoder, Interpretable ML, Cyber-Physical Systems**

I. Introduction

Anomaly detection in Cyber-Physical Systems (CPS) involves identifying unusual patterns or behaviors that stray from what we normally expect [1]. These anomalies can often indicate potential security issues, system failures, or challenges that may jeopardize critical infrastructure. For example, in water treatment plants, an unexpected change in flow or chemical levels could signal a cyber-attack or equipment malfunction that endangers the safety of the water supply. Therefore, it is crucial to design effective anomaly detection algorithms that can be deployed in real-time to detect these threats as they emerge.

Current approaches to anomaly detection in CPS leverage advanced machine learning techniques to learn complex, high-dimensional data. Machine learning methods such as Support Vector Machines (SVM), Random Forests, and k-nearest Neighbors have demonstrated promising results in detecting anomalies. Deep learning approaches have further advanced the field, with Long Short-Term Memory (LSTM) networks and autoencoders showing better performance in capturing temporal dependencies and intricate patterns in multivariate sensor data, making them particularly well-suited for CPS environments where numerous sensors continuously generate time-series data.

While deep learning models are effective for anomaly detection, they commonly lack interpretability. These ”black-box” models can detect anomalies but offer minimal explanation for why a specific event is flagged. This lack of clarity is especially concerning in the Cyber-Physical Systems (CPS) domain, where determining the cause of an anomaly is vital for quick action and correction. System operators require knowledge not only about the existence of an anomaly but also about which components are impacted and how they interact with the overall system performance. In safety-critical fields such as healthcare or autonomous vehicle systems, having interpretable outcomes is crucial for maintaining trust, ensuring proper human monitoring, and adhering to regulatory standards.

We propose using Sparse Autoencoders (SAEs)[2] for interpretable anomaly detection in CPS datasets. Unlike traditional autoencoders, SAEs impose sparsity in hidden layers, leading to a compact and meaningful latent representation. This enables the extraction of disentangled, important features corresponding to specific physical components or behaviors in the CPS. By analyzing SAE's features, such as model weights and latent code, we can gain insights into the importance of input features and map the predicted anomalies to specific physical components. This approach is evaluated on benchmark CPS datasets BATADAL[3], showing how sparse representations enhance detection accuracy and interpretability in real applications.

II. Related Work

A. Anomaly Detection

Anomaly detection identifies data points that deviate significantly from the expected pattern of a dataset. In supervised learning, algorithms such as Support Vector Machines (SVM) and Random Forest classifiers function by learning decision boundaries between normal and anomalous samples using labeled training data [4]. These methods achieve high accuracy when sufficient labeled examples exist, but their practical application is often limited by the scarcity of labeled anomalous instances in real-world scenarios. Self-supervised learning offers an alternative method that addresses this limitation by training exclusively on normal data samples. This approach constructs models that learn the intrinsic characteristics and statistical distributions of normal behavior, then identifies anomalies as samples that produce high reconstruction errors or low probability scores. Common self-supervised techniques include autoencoders that learn compressed representations of normal data, density estimation methods that model the

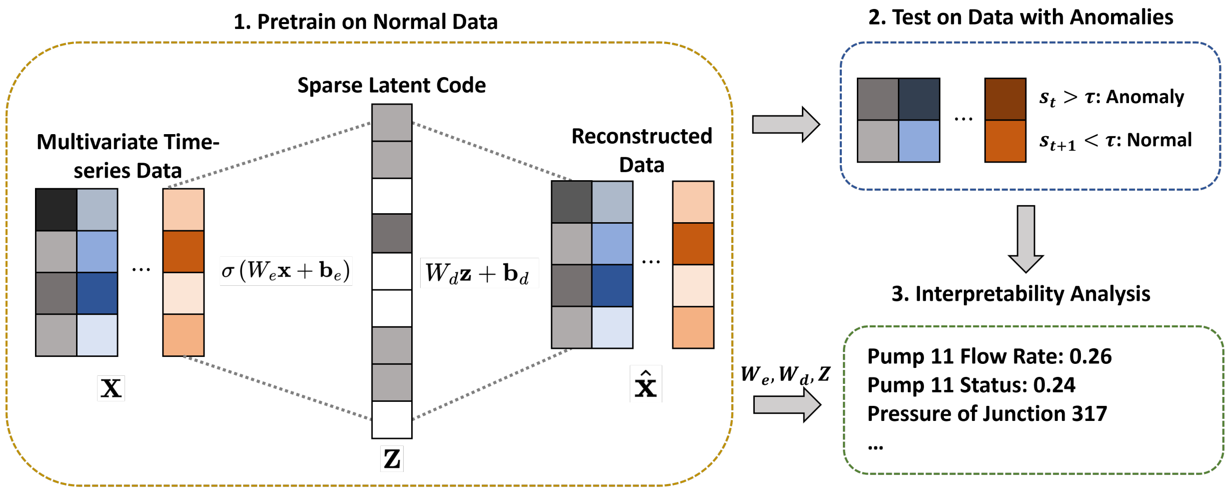

Fig. 1. The framework of interpretable CPS anomaly detection using SAEs. The process starts with pretraining the SAE on normal data, followed by testing on data containing anomalies. We analyze the input features that contribute most to the predicted anomalies to provide clear interpretations of the results.

probability distribution of normal data, and distance-based techniques that compute proximity to normal clusters [5].

B. Autoencoder-based CPS Anomaly Detection

Autoencoders have emerged as a powerful approach for anomaly detection in CPS environments. As described by Luo et al. [6], autoencoders learn compact representations of normal data by training on an encoding-decoding architecture that first compresses input data into a lower-dimensional latent space and then reconstructs the original input. Reconstruction error serves as an anomaly indicator during inference, with higher errors suggesting deviation from normal patterns. The RmsAnomaly model[7] advances this concept using convolutional autoencoders to capture both temporal dependencies and inter-sensor correlations through signature matrices and multi-scale windows, effectively analyzing different time scales. Further innovations include the MTS-DVGAN model [8], which combines deep generative models with contrastive learning, employing LSTM-based encoders to learn latent representations of multivariate time series data. To realize large-scale CPS, [9] proposed a decentralized approach using 1D Convolutional Autoencoders (1D-ConvAE) deployed directly on individual CPS components, enabling independent monitoring and faster anomaly detection.

C. SAEs for Interpretability

Sparse Autoencoders (SAEs)[2] are a variant of traditional autoencoders designed to enhance interpretability by enforcing sparsity constraints on latent activations. SAEs consist of an encoder that maps input data to a high-dimensional latent space and a decoder that reconstructs the input from these sparse activations. The encoder often uses techniques like L1 regularization to ensure only a small subset of latent units activate for any given input. This sparsity promotes disentangled representations, where individual latent units correspond to distinct, human-interpretable features. The interpretability of SAEs has been demonstrated across various domains. In image processing, sparse features often correspond to edge detectors and localized patterns. Recently, SAEs have gained significant attention for their role in interpreting neural representations in Large Language Models (LLMs)[10]. Researchers at Anthropic have shown that SAEs trained on LLM activations can identify interpretable features connected to specific concepts[11], sentiment patterns, and even deceptive behaviors in the model.

III. Overview

A. Problem Formulation

Let $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\} \in \mathbb{R}^{n\times d}$ represent a multivariate time series dataset from a CPS with n time points and d sensors. We define the anomaly detection task as learning a function $f : \mathbb{R}^d \to \mathbb{R}$ that maps each observation $\mathbf{x}_t$ to an anomaly score s_t, where higher values indicate greater likelihood of anomalous behaviors.

B. Solution Overview

Our project presents a framework for interpretable anomaly detection in cyber-physical systems (CPS) using sparse autoencoders (SAEs), as shown in Figure 1. The process consists of three main steps: First, we pretrain the SAE using only normal data to learn typical system patterns, transforming multivariate time-series data into sparse latent code and then reconstructing it. Second, we test the trained model on data containing anomalies to identify instances of abnormal behavior based on their anomaly scores. Third, we perform an interpretability analysis by examining the model weights and latent representations to identify which specific input features, such as pump flow rates or junction pressures, contribute most to the detected anomalies. This three-step approach not only detects anomalies but also provides operators with clear explanations of their possible causes, which is essential for effective response in CPS environments.

IV. Methods

A. ReLU Sparse Autoencoder

A Sparse Autoencoder consists of an encoder $E_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^h$ that maps input $\mathbf{x}$ to a latent representation $\mathbf{z} = E_\theta(x)$, and a decoder $D_\phi : \mathbb{R}^h \rightarrow \mathbb{R}^d$ that reconstructs the input as $\hat{\mathbf{x}} = D_\phi(\mathbf{z})$. Specifically, we can simply define:

$$\begin{aligned} \mathbf{z} &= \sigma\left(W_e \mathbf{x} + \mathbf{b}_e\right) \\ \hat{\mathbf{x}} &= W_d \mathbf{z} + \mathbf{b}_d \end{aligned} \tag{1}$$

where σ is the ReLU function, $W_e \in \mathbb{R}^{h\times d}$ and $W_d \in \mathbb{R}^{d\times h}$ are weight matrices, and $\mathbf{b_e} \in \mathbb{R}^h$ and $\mathbf{b}_d \in \mathbb{R}^d$ are bias vectors. I will modify the encoder and decoder architectures based on examples in II-B to better capture feature correlation and temporal characteristics in CPS datasets.

The SAE is trained to minimize two loss terms:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{sparsity}} \tag{2}$$

where:

- Reconstruction loss: $\mathcal{L}_{\text{recon}} = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$
- Sparsity loss: $\mathcal{L}_{\text{sparsity}} = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{z}_i\|_1$

B. BatchTopK SAE

However, this Vanilla ReLU SAE is outdated, and the BatchTopK SAE[12] is a better alternative. BatchTopK SAE retains only the top K activation values and zeros out the rest, meaning the k hyperparameter sets the desired sparsity without tuning the sparsity penalty λ in the Vanilla SAE:

$$\tilde{\mathbf{z}}_i = \operatorname{TopK}\left(\mathbf{z}_i, k\right). \tag{3}$$

The training objective simplifies to:

$$\mathcal{L}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - D_\phi\left(\tilde{\mathbf{z}}_i\right)\|_2^2 \tag{4}$$

This approach guarantees sparsity level regardless of the input distribution and improves feature disentanglement through competition between neurons. During inference, we can examine which specific neurons activate in response to an anomaly to gain insights into the contributing factors.

C. Anomaly Detection

1) Window-based Average Smoothing: When monitoring time-series data, it is crucial to consider temporal context for reliable anomaly detection. For each observation $\mathbf{x}_t$ at time t, we first calculate its reconstruction error:

$$e_t = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2 \tag{5}$$

To reduce the impact of transient errors and increase detection stability, we apply window-based average smoothing. This technique calculates the final anomaly score s_t by averaging reconstruction errors within a sliding window of size w:

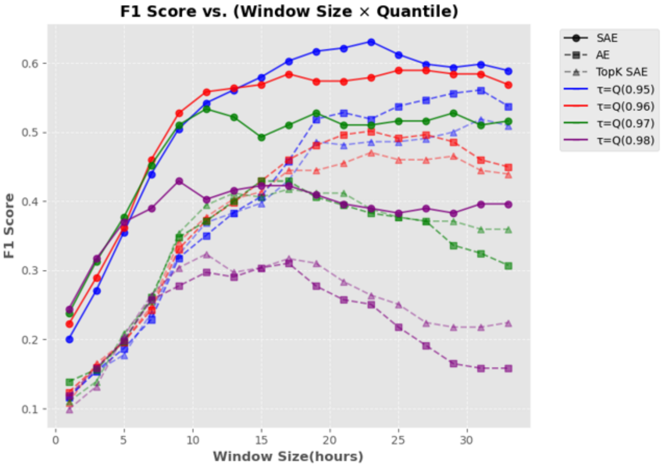

Fig. 2. F1 Score vs. Window Size for Different Models and Threshold Quantiles. The graph displays F1 scores (y-axis) against window size in hours (x-axis) for three models: ReLU SAE, AE, and TopK SAE. Four different threshold quantiles are shown. Window sizes range from 0 to 32 hours based on the window-based average smoothing technique.

$$s_t = \frac{1}{w} \sum_{i=0}^{w-1} e_{t-i} \tag{6}$$

This approach effectively incorporates temporal patterns and reduces false alarms by smoothing out isolated spikes in reconstruction error.

2) Defining Threshold: To classify observations as normal or anomalous, we establish a threshold τ where any observation with $s_t > \tau$ is considered anomalous. In our implementation, we determine τ based on the empirical distribution of test anomaly scores, typically setting it between the 95th and 99th percentile to balance detection sensitivity and false positive rate.

D. Interpretability Analysis

A key advantage of our ReLU SAE approach is its inherent interpretability through sparse feature activation patterns. For any input feature j, we quantify its contribution to anomaly detection using:

$$C_j = \sum_{k=1}^{h} \left|W_{kj}^e\right| \cdot z_k \cdot \left|W_{jk}^d\right| \tag{7}$$

This formula captures how each input feature influences the anomaly through the encoder weights W_{kj}^e, the activation of latent features z_k, and the decoder weights W_{jk}^d. By ranking these contribution scores, we identify the top-k features most responsible for detected anomalies, providing actionable insights for system operators.

V. Evaluation

In our experiments, we evaluate the performance of sparse autoencoders against baseline anomaly detection methods using the BATADAL (Battle of the Attack Detection Algorithms) dataset. This benchmark dataset provides an ideal testbed for our research, as it contains comprehensive data from the C-Town water distribution network, including both normal operations and simulated cyberattacks. The dataset is structured with a one-year training set that contains only normal operations, and a six-month test set that contains several labeled attacks that manipulate sensor readings and actuator states. These attacks represent realistic scenarios in which adversaries attempt to compromise the water system by altering water levels, flows, pressures, and the states of control equipment. BATADAL helps us see if autoencoders can catch small variations that might show a cyber attack on the water system. All models were run on an NVIDIA GTX 4090 GPU to ensure consistent computational performance across experiments. We select the F1 score as the metric because it provides a balanced assessment for the imbalance between normal operations and anomalous events in the BATADAL dataset.

We optimize the learning rate and latent code size for all models. Additionally, we optimize the sparsity bias λ for ReLU SAE and k for TopK SAE using the Optuna package[13].

A. Reconstruction Loss Analysis

In Figure 3 (a), we observe different training behaviors across the three autoencoder variants. AE achieves the lowest validation reconstruction loss with rapid convergence in early epochs, demonstrating efficient learning of normal data patterns. TopK SAE follows a similar convergence trajectory. In contrast, the ReLU SAE exhibits a notably slower convergence rate, with a smoother and more gradual decline throughout training. This suggests that while the sparsity loss objective creates a more challenging optimization landscape that requires more epochs to stabilize, it may avoid overfitting on the training and validation data.

Figure 3 (b) reveals important distinctions in how each model differentiates between normal and anomalous data. All three models successfully assign higher reconstruction loss to anomalies compared to normal samples, with some outliers in normal states potentially causing false alarms. While AE and TopK SAE display similar boxplot distributions, ReLU SAE demonstrates superior discriminative performance with two key advantages: a more compact box for normal states, indicating less variance in reconstruction error for normal data, and a higher median reconstruction loss for anomalies with clearer separation between distributions. This evidence suggests that despite ReLU SAE's slower training convergence, it ultimately creates more effective representations for distinguishing between normal operation and anomalies in test data.

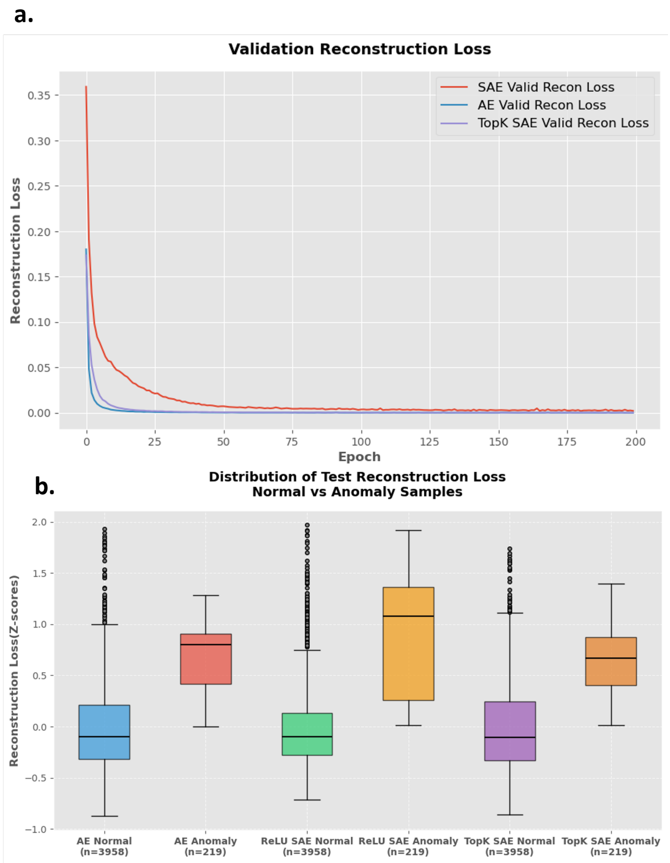

Fig. 3. (a) Validation reconstruction loss during training among all models. (b) Distribution of test reconstruction loss for normal and anomaly samples among all models.

B. Anomaly Detection Performance Analysis

Based on the figure 2, the ReLU SAE model consistently outperforms both standard AE and TopK SAE for anomaly detection across various window sizes and threshold settings. Performance for all models improves with increasing window size until approximately 20-25 hours, suggesting that temporal smoothing effectively reduces false alarms while maintaining detection capability. The threshold quantile also significantly impacts performance, with lower quantiles (0.95-0.96) generally yielding better F1 scores than higher ones (0.97-0.98). ReLU SAE shows particularly strong performance with $\tau = Q(0.95)$ and window sizes between 20-25 hours, achieving F1 scores above 0.62, while standard AE and TopK SAE peak around 0.55 and 0.51, respectively.

C. Interpretability Case Study

Figure 1 visualizes the detection trajectory of ReLU SAE across the test dataset timeline, showing its ability to identify most attack periods while completely missing the second attack period. In practical cyber-physical systems (CPS) deployments, detection systems must provide operators with explanations about which physical components may be compromised when anomalies are detected. To demonstrate this

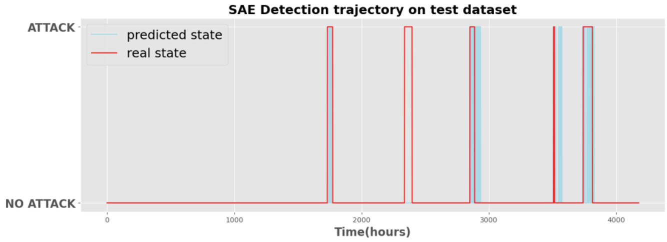

Fig. 4. Detection trajectory on test dataset. This figure compares the predicted attack states (blue line) against the actual attack states (red line) across the BATADAL test dataset timeline.

interpretability aspect, we will closely examine the first attack period successfully detected by the ReLU SAE model to extract meaningful explanations about the affected components and the nature of the attack, potentially revealing which physical components were manipulated during this security breach.

The first attack period lasted 50 hours, from September 13, 2016, at 23:00 to September 16, 2016, at 00:00. During this attack, the attacker modified the water level thresholds for tank T7 by altering SCADA transmissions to PLC9. This manipulation affected the control of pumps PU10 and PU11, resulting in low water levels in tank T7. The attack included a replay attack component specifically targeting tank T7 and featured 42 hours of SCADA concealment to hide the malicious activity from system operators.

We input the data from the first attack period into the trained ReLU SAE and derive the corresponding latent code. Then, we use Formula 7 to calculate the top 10 features with the highest contribution scores.

Figure V-C illustrates a comparative analysis of feature contribution patterns between ReLU SAE and standard AE models during anomaly detection for Attack 1. The ReLU SAE demonstrates superior interpretability by highlighting F_PU11 and S_PU11 as dominant contributors to the reconstruction error, with contribution scores of approximately 0.25 and 0.23, significantly higher than other features. This distribution aligns precisely with the attack description, which explicitly indicates PU11's involvement in manipulating tank T7's thresholds. This enables operators to trace the attack source directly to the compromised components. Conversely, while the standard AE successfully detects the same attack, its feature contribution profile reveals uniformly high scores (0.4-0.5) across multiple features (L_T3, P_J256, L_T6, etc.), failing to provide discriminative information about which components were actually targeted. This homogeneous distribution significantly diminishes the model's explanatory capacity, highlighting ReLU SAE's advantage in providing actionable intelligence for system operators responding to security breaches.

VI. Discussion

A. Sparsity Increases the Anomaly Detection Performance

The superior performance of ReLU SAE for anomaly detection can be attributed to its unique sparsity-inducing properties. Unlike standard AE, which may learn dense representations that capture both normal patterns and potential anomalies, ReLU SAE forces most activations to zero through the ReLU function while allowing only the most relevant features to remain active. This selective activation creates a more specialized latent space that effectively models normal behavior while being less capable of reconstructing anomalous patterns. When anomalies occur, ReLU SAE struggles to represent them accurately in its constrained sparse space, resulting in noticeably higher reconstruction errors compared to normal data.

B. Sparsity Values Important Input Features

The architectural differences between ReLU SAE and standard AE directly impact their contribution score distributions. ReLU SAE's non-linear activation and sparsity constraints create a selective latent representation that only activates neurons most relevant to the input data. This selectivity produces a skewed distribution of contribution scores where only the most anomalous features (F_PU11 and S_PU11) receive significantly higher weights, effectively filtering out less relevant signals. In contrast, standard AE's linear transformations, without sparsity constraints, distribute information more evenly across multiple latent dimensions, resulting in uniform contribution scores across features. While both models detect anomalies successfully, only ReLU SAE's sparsity mechanism provides the discriminative power to discover specific components responsible for the detected anomalies.

C. Increasing or Reducing the Latent Code Size?

When SAEs are applied to reconstruct the activation layer in large language models, they address the superposition[14] problem, wherein a single neuron can encode multiple distinct concepts simultaneously. To initialize monosemantic analysis in such contexts, researchers typically set the latent code size to be several times larger than the input dimension, facilitating the disentanglement of these overlapping representations. Conversely, in our CPS anomaly detection scenario, each input entry represents a singular, well-defined concept. Specifically, a discrete value recorded by a particular sensor within the system. This fundamental difference in data semantics requires an alternative architectural approach, where the latent code size should be intentionally constrained to be smaller than the input dimension. This design choice aligns with the classical autoencoder paradigm of dimensional reduction and information compression, forcing the network to learn a compact, efficient representation of the normal system state while simultaneously enhancing its sensitivity to anomalous patterns that deviate from this learned manifold.

D. Limitations

While our ReLU SAE demonstrates promising results for interpretable anomaly detection, several limitations should be acknowledged. First, this study represents a preliminary exploration of SAEs with best F1 scores of approximately 0.65, which is inferior to the reported SoTA's results, particularly

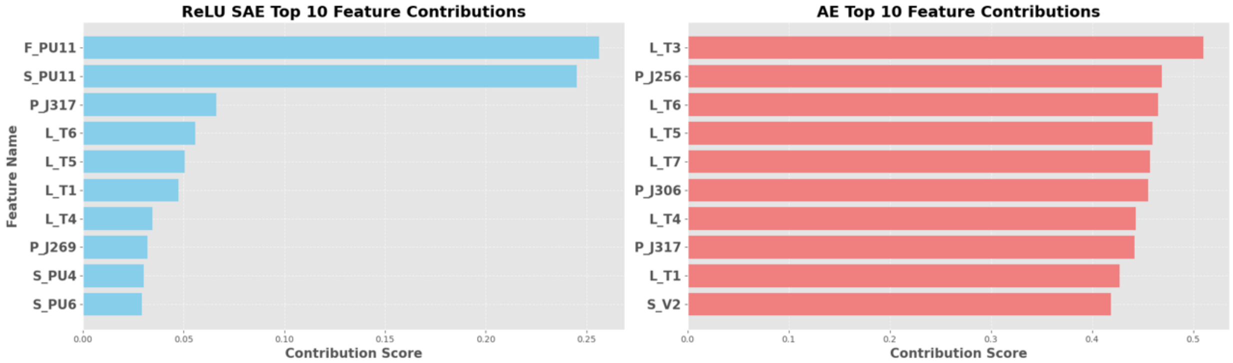

Fig. 5. Top 10 Features contribute to the high reconstruction loss in the first attack period, which ReLU SAE and AU both successfully detect. The contribution scores are calculated using Formula 7.

considering our model's complete failure to detect the second attack period in Figure 4. Future work can integrate these sparsity concepts into more advanced architectures, such as LSTM autoencoders, to better capture temporal dependencies in sequential data. Second, the effectiveness of our contribution score methodology requires broader validation across diverse anomaly scenarios and industrial systems. Although our case study shows promising alignment between identified components and predicted anomalies, additional interpretability studies with varied attack patterns and system configurations are necessary to establish the robustness and generalizability of this approach.

VII. Conclusion

This project focuses on interpretable anomaly detection for Cyber-Physical Systems using Sparse Autoencoders (SAEs). By imposing sparsity in the hidden layers, SAEs extract disentangled features corresponding to specific physical components that are responsible for the predicted anomalies. The approach is benchmarked on the BATADAL dataset and demonstrates better detection performance than the baseline. The interpretability case study validates that the sparsity in the latent code increases the interpretability of the model.

References

[1] D. Abshari and M. Sridhar, "A survey of anomaly detection in cyber-physical systems," *arXiv preprint arXiv:2502.13256*, 2025.

[2] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[3] R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, A. Ostfeld, D. G. Eliades, M. Aghashahi, R. Sundararajan, M. Pourahmadi, M. K. Banks *et al.*, "The battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks," *Journal of Water Resources Planning and Management*, vol. 144, no. 8, p. 04018048, 2018.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[5] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[6] Y. Luo, Y. Xiao, L. Cheng, G. Peng, and D. Yao, "Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021.

[7] Z. Dong, K. Liu, D. Han, Y. Cao, and Y. Xia, "Reconstruction-based multi-scale anomaly detection for cyber-physical systems," in *2022 4th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2022, pp. 1–6.

[8] H. Sun, Y. Huang, L. Han, C. Fu, H. Liu, and X. Long, "Mts-dvgan: Anomaly detection in cyber-physical systems using a dual variational generative adversarial network," *Computers & Security*, vol. 139, p. 103570, 2024.

[9] C. Goetz and B. Humm, "Decentralized real-time anomaly detection in cyber-physical production systems under industry constraints," *Sensors*, vol. 23, no. 9, p. 4207, 2023.

[10] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," *arXiv preprint arXiv:2309.08600*, 2023.

[11] N. Anthropic, A. Conmy, N. Joseph, J. Kornblith, L. Kravec, and B. Shlegeris, "Towards monosemanticity: Decomposing language models with dictionary learning," *arXiv preprint arXiv:2402.03208*, 2024.

[12] B. Bussmann, P. Leask, and N. Nanda, "Batchtopk sparse autoencoders," *arXiv preprint arXiv:2412.06410*, 2024.

[13] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[14] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen *et al.*, "Toy models of superposition," *arXiv preprint arXiv:2209.10652*, 2022.