

Article

TCN-AE with CUSUM Control Chart for Online Anomaly Detection in Hydraulic Support Pressure Data

Cong Wang ^{1,†} , Wei Xin ^{1,*} , Jun Li ^{1,*} , Xigui Zheng ^{1,2} , Yu Zhao ^{1,3} and Zhongguo He ³

¹ School of Mines, China University of Mining and Technology, Xuzhou 221116, China; smart.wang@cumt.edu.cn (C.W.); ckzxcg@cumt.edu.cn (X.Z.); ds23020235p32dk@cumt.edu.cn (Y.Z.)

² School of Mines and Mechanical Engineering, Liupanshui Normal University, Liupanshui 553004, China

³ Guizhou Kailin Group Co., Ltd., Guiyang 550300, China

* Correspondence: tb24020050a41ky@cumt.edu.cn (W.X.); lb24020006@cumt.edu.cn (J.L.)

† These authors contributed equally to this work.

Abstract

Hydraulic supports in coal mining faces require continuous pressure monitoring to detect anomalies indicative of roof instability or equipment failure. Existing reconstruction-based methods rely on standard convolutional or recurrent encoders whose limited receptive fields or coarse temporal representations restrict detection accuracy; static per-window thresholding further discards temporal continuity during online deployment. This study proposes a temporal convolutional network autoencoder (TCN-AE) coupled with a Cumulative Sum (CUSUM) control chart for online anomaly detection in hydraulic support pressure data. The TCN encoder uses dilated convolutions with symmetric padding and residual connections, producing an exponentially expanding receptive field that captures temporal patterns at multiple scales. The CUSUM chart accumulates sustained positive deviations in the reconstruction error sequence, improving detection sensitivity while suppressing isolated false alarms. Component analysis experiments on synthetic anomalies show TCN-AE achieves an AUC of 0.811, outperforming CNN, LSTM, GRU, and fully connected autoencoder variants, along with Isolation Forest and One-Class SVM. On a manually curated real fault test set, where per-window reconstruction scores carry negligible discriminative information (AUC = 0.586, near chance), the CUSUM strategy exploits temporal continuity to improve F1 from 0.213 to 0.905 for TCN-AE. This +0.692 gain is driven entirely by temporal accumulation rather than model discriminability, demonstrating that the CUSUM framework is most valuable precisely when per-window signals are weakest.

Keywords: temporal convolutional network; autoencoder reconstruction; CUSUM control chart; hydraulic support monitoring; unsupervised anomaly detection

MSC: 68T07



Academic Editor: Renato Bruni

Received: 17 May 2026

Revised: 17 June 2026

Accepted: 18 June 2026

Published: 26 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Hydraulic supports are the main load-bearing structures in fully mechanized coal mining faces, maintaining strata stability and preventing roof collapse [1,2]. Their structural integrity and operational reliability directly affect underground safety and production continuity [3,4]. Modern hydraulic supports are instrumented with pressure sensors that continuously record column pressure at minute-level intervals, generating large volumes of time-series data encoding the dynamic interaction between the support and the overlying strata [5]. Real-time anomaly detection in these pressure data—deviations from normal

patterns signaling roof instability, support failure, or hydraulic leakage—requires methods that operate on continuous streams without labeled fault examples.

Reconstruction-based autoencoders are widely used for unsupervised time-series anomaly detection across industrial domains [6,7]. An autoencoder trained on normal data reconstructs normal patterns accurately, while anomalous inputs yield elevated reconstruction error. This approach has been applied to fault diagnosis in hydraulic pumps [8,9], rotating machinery [10,11], and industrial monitoring [12,13], often matching or exceeding supervised methods when labeled fault data are scarce [14,15].

Within hydraulic support monitoring, Zheng et al. [16] combined convolutional autoencoders (CNN-AE) with dynamic time warping (DTW) for anomaly detection in mining hydraulic support pressure data. Park et al. [17] evaluated CNN and LSTM for detecting abnormal pulsating pressures in hydraulic accumulators, finding that deep models outperformed threshold-based and conventional machine learning methods. Neufeld and Schmid [18] showed that early anomaly identification in hydraulic test systems can prevent cascading failures, and Kim and Heo [5] demonstrated that machine-learning-based feature extraction supports condition monitoring across multiple hydraulic applications. Beyond hydraulic systems, convolutional architectures are common in industrial time-series anomaly detection [19,20], and hybrid CNN-RNN or CNN-attention models have reported additional improvements [21–23]. Other architectures applied to related fault diagnosis tasks include deep belief networks [24], LSTM-based autoencoders [25], and multirate deep learning models [26].

Two limitations constrain existing methods. First, the CNN encoders used in current hydraulic support anomaly detection rely on local convolution kernels whose receptive fields grow only linearly with depth, restricting the capture of long-range temporal dependencies in mining-cycle pressure sequences [27,28]. Recurrent architectures such as LSTM and GRU can model longer sequences through gated hidden states [29,30], but their global hidden-state representation introduces considerable reconstruction error on fine-grained temporal patterns, as confirmed by the two-tier reconstruction precision gap in our component analysis experiments. Second, existing approaches adopt static thresholding on per-window reconstruction scores to flag anomalies, treating each test sample independently without exploiting the temporal structure of the score sequence itself. In online deployment, where extended normal operation precedes anomaly onset, this per-sample decision discards contextual information useful for suppressing transient noise while accumulating evidence for sustained anomalies.

This study proposes a temporal convolutional network autoencoder (TCN-AE) coupled with a CUSUM control chart for online anomaly detection in hydraulic support pressure data [31,32]. The combination of deep reconstruction models with sequential detection statistics follows the NN-CUSUM framework proposed by Gong et al. [33], who established a general theoretical condition for using neural network outputs as CUSUM statistics in online change-point detection. The TCN encoder uses dilated convolutions with symmetric padding and residual connections, producing an exponentially expanding receptive field that captures temporal patterns at multiple scales while keeping the parameter count low. The CUSUM control chart is applied to the raw reconstruction error sequence as a post-processing module: it accumulates sustained positive deviations above the in-control baseline and resets during normal operation, exploiting deployment continuity to improve detection sensitivity and suppress isolated false alarms.

This study proposes a temporal convolutional network autoencoder (TCN-AE) coupled with a CUSUM control chart for online anomaly detection in hydraulic support pressure data [31,32]. The TCN encoder uses dilated convolutions with symmetric padding and residual connections, producing an exponentially expanding receptive field that captures

temporal patterns at multiple scales while keeping the parameter count low. The CUSUM control chart is applied to the raw reconstruction error sequence as a post-processing module: it accumulates sustained positive deviations above the in-control baseline and resets during normal operation, exploiting deployment continuity to improve detection sensitivity and suppress isolated false alarms.

Component analysis experiments show that the dilated convolution architecture with residual connections achieves a validation MSE on the order of 10^{-4} and an AUC of 0.811, outperforming CNN, LSTM, GRU, and fully connected autoencoders as well as Isolation Forest and One-Class SVM baselines. The CUSUM-based dynamic threshold accumulates sustained deviations across consecutive anomalous windows, improving F1 over static thresholding by +0.159 on synthetic anomalies and +0.692 on real faults for TCN-AE; the gain is largest when per-window discriminability is weakest. A manually curated real fault test set is constructed from naturally occurring anomaly state records through a data-driven filtering pipeline to supplement synthetic anomaly evaluation. Results on this set reveal a notable gap between synthetic and real anomaly detection difficulty.

2. Methodology

2.1. Problem Definition

This study formulates anomaly detection as an unsupervised reconstruction problem: an autoencoder is trained exclusively on normal pressure sequences, and anomalies are identified by their elevated reconstruction error.

Formally, let $\mathbf{x} = [x_1, x_2, \dots, x_T] \in \mathbb{R}^T$ denote a normalized pressure window of length $T = 720$. The autoencoder learns a mapping $f: \mathbb{R}^T \rightarrow \mathbb{R}^T$ that minimizes the reconstruction loss:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T (x_t - \hat{x}_t)^2 \quad (1)$$

where $\hat{x}_t = f(\mathbf{x})_t$ is the reconstructed value. An anomaly score $s(\mathbf{x})$ is defined as the MSE between input and reconstruction:

$$s(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T (x_t - \hat{x}_t)^2 \quad (2)$$

Two modifications to the conventional AE framework are introduced. First, the encoder is replaced by a TCN whose dilated convolutions with symmetric padding produce an exponentially growing receptive field, capturing long-range temporal dependencies that standard CNN encoders miss. Second, a Cumulative Sum (CUSUM) control chart is applied to the raw MSE scores to exploit temporal context in the score sequence, accumulating sustained positive deviations above the in-control baseline while resetting automatically during normal operation. Figure 1 illustrates the overall pipeline.

The term ‘online’ in this work denotes that decisions are made continuously as new windows become available, with the detection model already trained. Each window of $T = 720$ min (12 h) produces one reconstruction score and one CUSUM-based decision. Although a 12 h observation before each decision introduces inherent latency, this is within the operational tolerance of hydraulic support monitoring, where faults develop over hours to days. The CUSUM statistic accumulates information across consecutive windows, enabling detection within 3–5 windows (36–60 h) of anomaly onset, as shown in the multi-segment evaluation (Section 3.3.3).

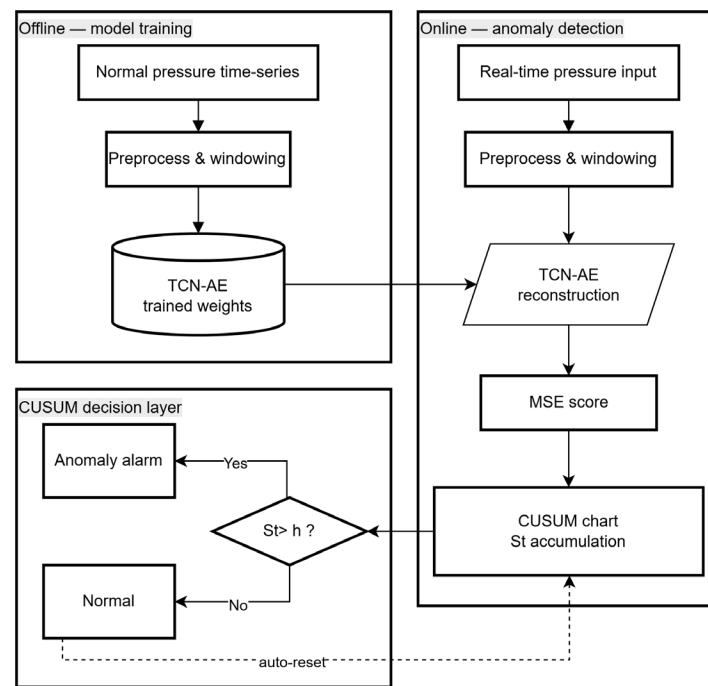


Figure 1. Overview of the proposed TCN-AE anomaly detection pipeline. Raw pressure data are preprocessed and segmented into sliding windows, which are fed into the TCN-AE trained exclusively on normal sequences. The reconstruction MSE is processed by a CUSUM control chart and compared against a decision interval threshold to produce the final normal/anomaly decision.

2.2. TCN-AE Architecture

2.2.1. Temporal Convolutional Network Encoder

Standard CNN-based autoencoders employ regular convolutions whose receptive fields grow only linearly with depth, limiting their ability to model long-range dependencies. To address this, we adopt a TCN as the encoder, which stacks dilated convolutions with symmetric padding to achieve an exponentially expanding receptive field while maintaining a linear parameter count.

Each convolutional layer uses symmetric (equal left-right) padding of $(K - 1) \cdot d/2$, where K is the kernel size and d is the dilation factor, preserving the temporal dimension without causal restriction:

$$y[t] = \sum_{k=0}^{K-1} w[k] \cdot x \left[t + \left(k - \left\lfloor \frac{K}{2} \right\rfloor \right) \cdot d \right] \quad (3)$$

where K is the kernel size, $w[k]$ are the learnable weights, and d is the dilation factor.

The dilation factor increases geometrically across layers as $d = 2^l$ at layer l , so the receptive field expands exponentially with depth while the parameter count remains linear. For a network of L layers and kernel size K , the effective receptive field is:

$$RF = 1 + \sum_{l=0}^{L-1} (K - 1) \cdot 2^l \quad (4)$$

Each TCN layer is realized as a residual block (Figure 2) comprising two dilated convolutions with symmetric padding; the first is followed by ReLU activation and spatial dropout, while the second is followed only by spatial dropout before the residual addition:

$$\text{Block}_l(\mathbf{x}) = \text{ReLU}(\text{Conv}_{d=2^l}(\text{ReLU}(\text{Conv}_{d=2^l}(\mathbf{x}))) + \mathbf{x}) \quad (5)$$

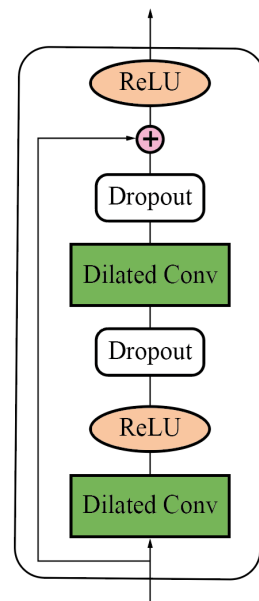


Figure 2. Internal structure of a single TCN residual block at layer l . The main path consists of two dilated convolutions with symmetric padding ($d = 2^l$, $K = 3$); the first is followed by ReLU activation and dropout, and the second by dropout only. The skip connection preserves the original input via identity mapping (or 1×1 convolution when channel dimensions mismatch), and the two paths are combined through element-wise addition followed by a final ReLU.

A 1×1 convolution is applied on the skip branch when the input and output channel dimensions differ.

The TCN-AE uses $L = 3$ layers, initial channel width 32, double-growth strategy (channels: 32, 64, 128), kernel size $K = 3$, bottleneck dimension 4, and dropout 0.1.

2.2.2. Encoder–Decoder Structure

The full TCN-AE architecture (Figure 3) consists of three stages. In the first stage, the input tensor ($B, 1, 720$) passes through three TCN residual blocks at full temporal resolution (stride = 1), producing a feature map ($B, 128, 720$) with RF = 15. In the second stage, two strided Conv1d layers (stride = 2) downsample the temporal axis from 720 to 180, and a 1×1 convolution compresses the 128-channel representation to a 4-channel bottleneck ($B, 4, 180$). The third stage mirrors the encoder via transposed convolutions, upsampling the bottleneck back to ($B, 1, 720$).

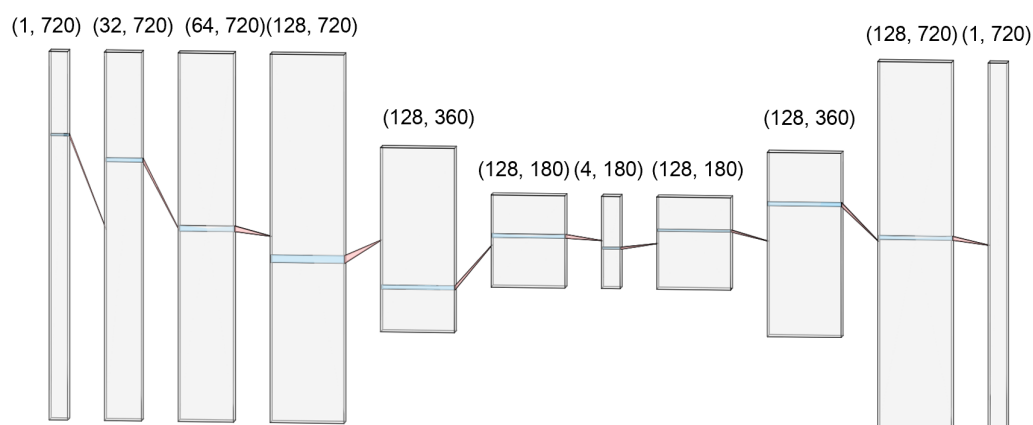


Figure 3. TCN-AE architecture. The encoder consists of three TCN residual blocks (dilations $d \in \{1, 2, 4\}$, channels $\{32, 64, 128\}$) operating at full temporal resolution, followed by two strided

convolutional downsampling layers ($720 \rightarrow 180$) and a 1×1 bottleneck compression to 4 channels. The decoder mirrors this structure via transposed convolutions, restoring the original temporal resolution. The MSE between input and reconstruction serves as the anomaly score.

The bottleneck compresses normal pressure dynamics into a compact latent representation. Because the model is trained exclusively on normal data, it learns to reconstruct normal patterns with high fidelity but cannot replicate anomalous deviations.

2.2.3. Comparison with CNN-AE Baseline

The CNN-AE baseline uses a standard convolutional autoencoder with comparable encoder–decoder depth and the same bottleneck dimension (4 channels) [6]. Its encoder employs local convolution kernels ($k = 7$) with an effective receptive field of approximately 20 steps, without dilation or residual connections.

2.3. CUSUM Dynamic Threshold Strategy

2.3.1. Motivation

Conventional static thresholding sets a fixed decision boundary $\tau = \mu + k\sigma$, where μ and σ are the mean and standard deviation of the validation MSE distribution. Each test sample is classified independently, ignoring temporal context in the score sequence. In online monitoring, where extended normal operation precedes anomaly onset, exploiting this continuity can improve detection performance.

2.3.2. CUSUM Control Chart

A one-sided CUSUM statistic accumulates sustained positive deviations in the raw MSE score sequence:

$$S_t = \max(0, S_{t-1} + (s(x_t) - \mu_0) - \delta) \quad (6)$$

where $S_0 = 0$, μ_0 is the in-control mean estimated from the validation MSE of normal windows, and δ is the allowance parameter absorbing small random fluctuations around the baseline. An alarm is raised when $S_t > h$, where h is the decision interval.

The CUSUM statistic has two properties suited to online detection. First, automatic reset: when $s(x_t) \leq \mu_0 + \delta$, the increment is non-positive and S_t decays toward zero via the $\max(0, \cdot)$ operator, preventing residual inertia from contaminating subsequent normal samples, a failure mode of alternative smoothing approaches discussed in Section 4.3. Second, persistent-deviation sensitivity: isolated spikes contribute only a single increment to S_t and are unlikely to exceed h unless very large; sustained anomalies accumulate increments and drive S_t past h even when individual samples are only moderately elevated.

We set $\delta = 0.5\sigma_0$ and $h = 5.0\sigma_0$, where σ_0 is the standard deviation of the validation normal MSE scores. These values follow standard statistical process control guidelines: $\delta = 0.5\sigma$ balances sensitivity and noise tolerance, while $h = 5\sigma$ ensures a low false-alarm rate during normal operation.

2.3.3. Percentile-Based Control Limit for Static Threshold

For the static threshold baseline, we use a percentile-based control limit applied directly to raw MSE scores:

$$\tau = P_{95}(\mathbf{s}_{\text{val}}) + \alpha \cdot \sigma_{\text{val}} \quad (7)$$

where $P_{95}(\mathbf{s}_{\text{val}})$ is the 95th percentile of validation MSE scores, σ_{val} is the validation MSE standard deviation, and $\alpha = 2.0$. Anchoring on the 95th percentile rather than the mean provides robustness against heavy-tailed MSE distributions produced by recurrent models, whose occasional reconstruction outliers inflate the mean but leave the upper percentile largely unaffected.

2.4. Training Protocol

All autoencoder models minimize the MSE reconstruction loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{BT} \sum_{i=1}^B \sum_{t=1}^T \left(x_t^{(i)} - \hat{x}_t^{(i)} \right)^2 \quad (8)$$

Optimization is carried out with the Adam optimizer [34]. All models use a learning rate of $\eta = 10^{-3}$ and weight decay 10^{-5} , and are trained for a fixed 300 epochs, saving the checkpoint with the lowest validation loss.

Raw pressure values are linearly rescaled to $[0, 1]$ using MinMax normalization with statistics computed from the training set alone:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (9)$$

where $x_{\min}$ and $x_{\max}$ are the training-set extrema. Normalization parameters are derived exclusively from the training partition to prevent information leakage.

All deep learning models share the same training hyperparameters (Adam, lr = 1×10^{-3} , 300 epochs, batch size 64) to ensure comparability. Validation MSE values are as follows: TCN-AE 2.72×10^{-5} , CNN-AE 1.60×10^{-4} , LSTM-AE 2.25×10^{-3} , and GRU-AE 2.84×10^{-3} . A grid search over learning rates $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}\}$ and hidden dimensions $\{32, 64, 128\}$ confirmed that the default configuration achieves the lowest validation loss for both LSTM-AE and GRU-AE, ruling out under-tuning as the cause of the gap.

2.5. Baseline Models

TCN-AE is compared against six baselines spanning deep learning and traditional anomaly detection methods. CNN-AE shares the same autoencoder framework but uses standard convolutional encoding [28]. LSTM-AE and GRU-AE use recurrent encoders with comparable parameter counts and symmetric decoder structures [35]. Vanilla AE is a fully connected autoencoder with similar encoder–decoder depth [6]. Isolation Forest [36] and One-Class SVM [37] operate on flattened 720-dimensional feature vectors and represent non-reconstruction-based approaches.

All deep learning models follow the same training protocol.

3. Experiments

Throughout this section, two distinct F1-score reporting protocols are used. The component contribution analysis (Section 3.3) and CUSUM evaluation (Sections 3.6 and 4.3) report F1 under fixed thresholds calibrated on the validation set (static percentile-based or CUSUM decision interval). The model comparison (Section 3.4) reports F1 at the validation-calibrated P99 threshold (the 99th percentile of validation MSE, controlling $\leq 1\%$ false-positive rate on normal data), supplemented by threshold-independent AUC-ROC as the primary ranking metric. These two protocols serve distinct purposes: the former evaluates deployment-realistic performance under a predetermined decision rule, while the latter characterizes each model's best achievable discriminability independent of any specific threshold choice.

3.1. Dataset

3.1.1. Data Source

The experimental data are collected from hydraulic supports operating in a coal mining face, with pressure sensors recording at 1 min intervals. The dataset comprises 1,767,951 records spanning multiple working cycles. Each record includes a timestamp,

column pressure, and the support's current state—setting, yielding, advancing, maintenance, or anomaly. The anomaly state indicates periods where the support status cannot be reliably determined from sensor readings.

Figure 4 illustrates characteristic mining-cycle pressure patterns: each cycle begins with a rapid pressure rise, transitions to a sustained high-pressure phase, and ends with a gradual decay. Cycle duration varies with mining pace and geological conditions.

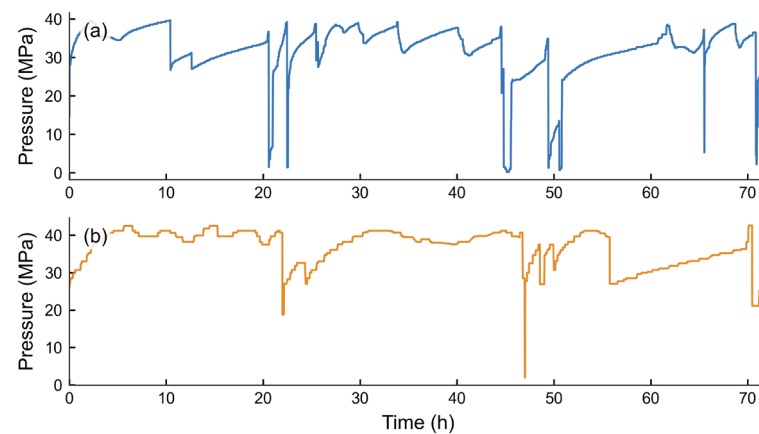


Figure 4. (a,b) Representative pressure time series from two hydraulic supports. Each mining cycle exhibits a rapid pressure rise during advancement, a sustained high-pressure phase, and a gradual decay. Signals are sampled at 1 min intervals.

3.1.2. Preprocessing

Records in the anomaly state are discarded. The remaining column pressure values are linearly rescaled to $[0, 1]$ using MinMax normalization with training-set extrema. The data are partitioned chronologically into training (80%), validation (10%), and test (10%) sets. Sliding windows of length $T = 720$ are then extracted independently within each partition with a stride of $\Delta = 72$, ensuring no window straddles partition boundaries. The training set is restricted to normal windows, identified by retaining windows from active mining phases where the support state is not an anomaly and the pressure signal exhibits characteristic cyclic patterns.

3.1.3. Synthetic Anomaly Injection

Since near-constant signals are trivially reconstructed and records in the anomaly state are unsuitable for evaluation, we adopt a controlled synthetic anomaly injection strategy. Five anomaly types are defined to cover the range of fault patterns observed in hydraulic support monitoring: spike, offset, noise, scale, and drift. Although five types are injected, the detection task remains strictly binary (normal vs. anomalous); type-specific recall analysis (in Section 3.4) characterizes detection difficulty only, not multi-class classification capability. The amplitude parameters (Table 1) are calibrated to produce moderate, physically plausible deviations that challenge the detection pipeline without being trivially detectable.

Table 1. Synthetic Anomaly Types and Parameters.

Anomaly Type	Description	Parameter Range	Affected Region
Spike	Point peaks at random positions	Amplitude: $[0.15, 0.40]$	3% positions
Offset	Constant shift on a random segment	Offset: $\pm[0.06, 0.15]$	15% of the window
Noise	Localized Gaussian noise	σ : $[0.02, 0.06]$	40% of the window
Scale	Amplitude scaling on a random segment	Scale: $[1.15, 1.60]$	10% of the window
Drift	Linear drift on a random segment	Range: $\pm[0.06, 0.15]$	20% of the window

Each anomalous test sample receives one anomaly type (probability 0.80) or two combined types (probability 0.20). The test set comprises 368 samples balanced at a 50% anomaly ratio—187 normal and 187 anomalous—and is randomly shuffled to eliminate temporal ordering bias.

Figure 5 demonstrates the visual appearance of each anomaly type and the corresponding injection error magnitude. Spike anomalies produce sharp, localized injection errors at isolated points. Offset and drift anomalies generate sustained, uniformly elevated error across the affected segment. Scale anomalies produce an error proportional to the local signal amplitude. Noise anomalies create a distributed, low-amplitude error pattern that most closely resembles natural signal fluctuations.

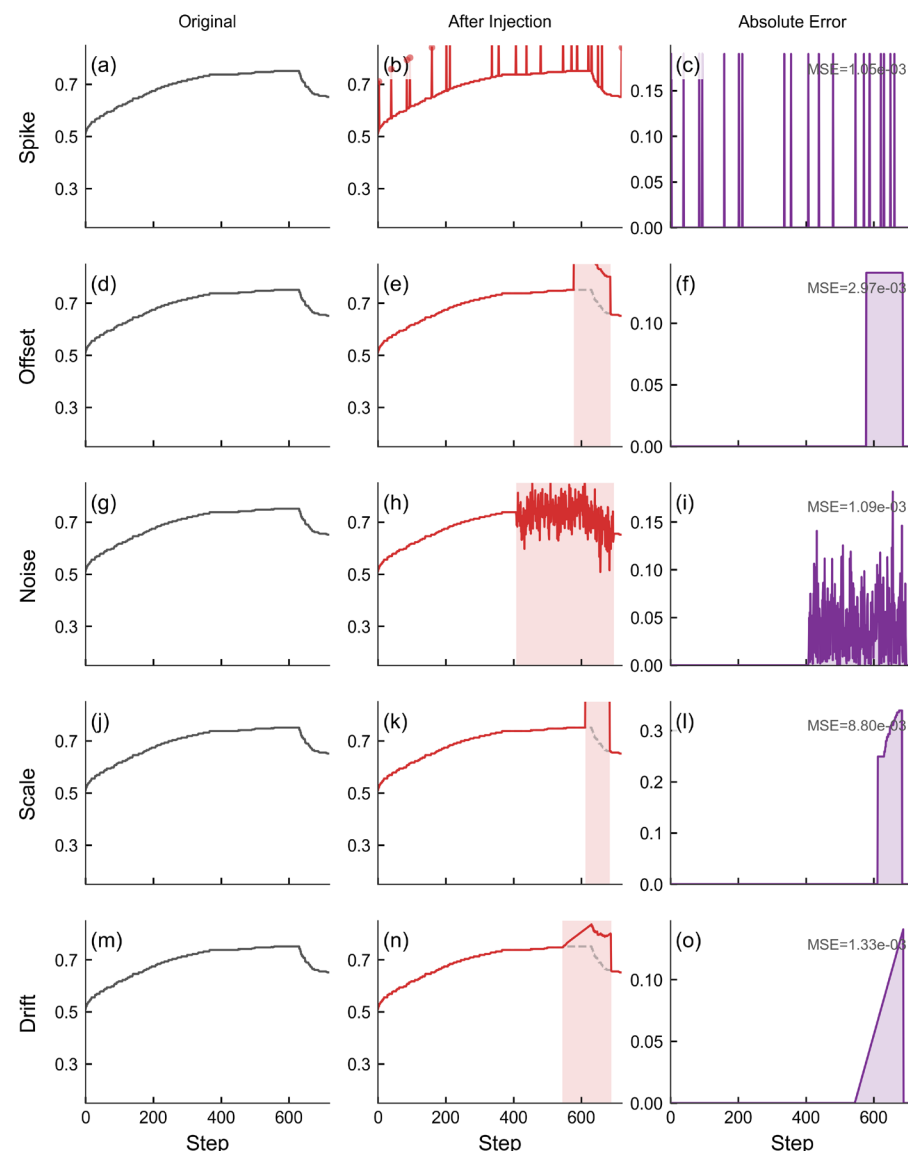


Figure 5. (a–o) Examples of the five synthetic anomaly types injected into normal pressure windows. For each row: original signal (gray), anomaly injected signal (red) with affected region highlighted, and absolute injection error $\|x_{inj} - x_{orig}\|$ (purple). The anomaly parameters correspond to the moderate amplitudes defined in Table 1.

3.2. Evaluation Metrics

Two complementary evaluation protocols are adopted. The ablation study (TCN-AE vs. CNN-AE $\times$ CUSUM vs. static threshold) is assessed using precision, recall, F1-score, and AUC-ROC under the specific threshold method. The model comparison uses AUC-ROC as

the primary, threshold-independent metric of overall discriminability, supplemented by F1-Score at the validation-calibrated P99 threshold, which controls the expected false-positive rate on normal data to $\leq 1\%$ without exposure to test labels.

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2PR}{P + R} \quad (10)$$

The dual-metric protocol is motivated by the wide variance in reconstruction precision across models. Models with low reconstruction fidelity (LSTM-AE, GRU-AE) exhibit high baseline MSE variance; applying a single fixed control limit would suppress their F1 to near zero regardless of any genuine discriminability, masking the true performance differences captured by AUC.

3.3. Component Contribution Analysis

We conduct a 2×2 factorial component analysis (TCN vs. CNN encoder $\times$ CUSUM vs. static threshold) to isolate the contribution of each proposed component. Both threshold methods use the validation MSE distribution as the basis for their decision boundaries: the static threshold applies a fixed percentile-based control limit directly to raw MSE, while CUSUM accumulates positive deviations above the in-control mean with an allowance of $\delta = 0.5\sigma_0$ and a decision interval of $h = 5.0\sigma_0$. To evaluate CUSUM in its intended online deployment scenario, where extended normal operation precedes anomaly onset, the test windows are arranged in deployment order: all normal windows (sorted by ascending MSE), followed by all anomalous windows (sorted by ascending MSE). Figure 6 summarizes the component contribution analysis across all four configurations.

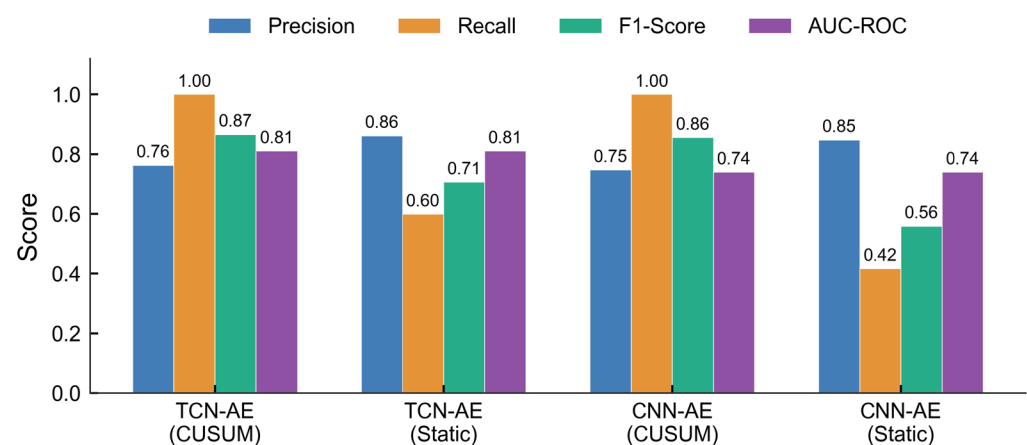


Figure 6. Component contribution analysis TCN-CNN encoders with CUSUM/static threshold strategies under online deployment ordering.

3.3.1. Contribution of TCN Encoder

Under both threshold strategies, TCN-AE outperforms CNN-AE in AUC-ROC (0.811 vs. 0.740), indicating that the dilated non-causal convolution architecture with residual connections captures a richer discriminative representation than the local CNN encoder.

3.3.2. Contribution of CUSUM Threshold Strategy

Under the online deployment ordering, CUSUM improves recall for both models compared to static thresholding. For TCN-AE, CUSUM achieves perfect recall (1.000, up from 0.599) at the cost of a precision decrease from 0.862 to 0.763, yielding an F1 gain from 0.707 to 0.866 ($\Delta F_1 = +0.159$). For CNN-AE, CUSUM also achieves perfect recall (1.000, up from 0.417), with precision dropping from 0.848 to 0.748, resulting in a net F1 improvement from 0.559 to 0.856 ($\Delta F_1 = +0.297$).

TCN-AE achieves the highest F1 under CUSUM (0.866); its lower reconstruction error variance produces tighter CUSUM behavior on the same relative parameters, giving a more balanced precision–recall profile ($P = 0.763$, $R = 1.000$) than CNN-AE ($P = 0.748$, $R = 1.000$). Both models achieve perfect recall under CUSUM by accumulating sustained deviations across consecutive anomalous windows, which necessarily trades precision for recall.

The precision–recall trade-off is consistent with the AUC difference between the two models (0.811 vs. 0.740). Because AUC depends only on the rank ordering of reconstruction scores, it is invariant to the post-processing strategy and reflects the model’s inherent discriminability. Both threshold strategies produce identical AUC for each model.

3.3.3. Multi-Segment Online Evaluation

The results in Table 2 assume a single normal-to-anomaly transition in deployment ordering. To evaluate CUSUM under more realistic streaming conditions with multiple fault–normal–fault cycles, we construct a four-segment alternating test stream (each segment: 46 normal windows followed by 46 anomalous windows; 368 total windows), with the CUSUM statistic resetting independently between adjacent anomalous blocks. Under this multi-segment scenario, TCN-AE + CUSUM achieves $F1 = 0.723$, lower than the single-segment F1 of 0.866 reported in Table 2. This decrease is expected: each segment contains fewer windows for CUSUM to accumulate evidence, and the reset between segments discards the cumulative advantage built during normal operation. The per-window false-alarm rate is 0.277 (stable across all four segments), and the average time-to-detection from the first anomalous window is 2.7 windows. Under the same deployment ordering, CUSUM marginally outperforms single-segment static thresholding ($F1 = 0.707$), confirming that the CUSUM mechanism provides meaningful detection capability beyond single-segment ordering.

Table 2. Component Contribution Analysis Results (online deployment order: normal $\rightarrow$ anomaly).

Encoder	Threshold	Precision	Recall	F1-Score	AUC-ROC
TCN-AE	CUSUM	0.763	1.000	0.866	0.811
TCN-AE	Static	0.862	0.599	0.707	0.811
CNN-AE	CUSUM	0.748	1.000	0.856	0.740
CNN-AE	Static	0.848	0.417	0.559	0.740

3.4. Comparison with Baseline Models

Table 3 compares all seven models using AUC-ROC (primary) and F1-Score at the validation-calibrated P99 threshold (secondary). AUC is threshold-independent; the P99 threshold controls the expected false-positive rate on normal data to $\leq 1\%$ without exposure to test labels.

Table 3. Model Comparison Results. Deep learning models (top) and traditional machine learning methods (bottom) are separated by a horizontal rule.

Model	Precision	Recall	F1	AUC
TCN-AE	0.867	0.594	0.705	0.811
CNN-AE	0.851	0.396	0.540	0.740
LSTM-AE	0.520	0.070	0.123	0.680
GRU-AE	0.552	0.086	0.148	0.659
Vanilla-AE	0.471	0.043	0.078	0.665
One-Class SVM	0.357	0.027	0.050	0.569
Isolation Forest	0.550	0.059	0.106	0.550

Figure 7 visualizes the multi-metric comparison, Figure 8 presents the ROC curves with AUC annotations, and Figure 9 shows the confusion matrices at each model's optimal F1 threshold.

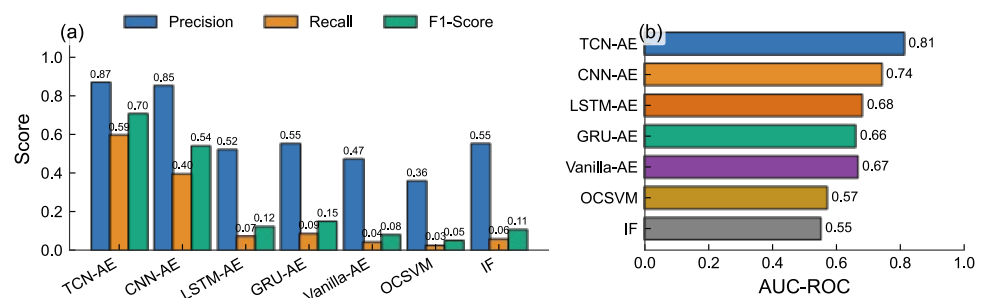


Figure 7. Comprehensive comparison of all models. (a) grouped bar chart of precision, recall, and F1-score at the validation-calibrated P99 threshold for all seven models. (b) horizontal bar chart of AUC-ROC. TCN-AE achieves the best overall performance (AUC 0.811, F1 0.705).

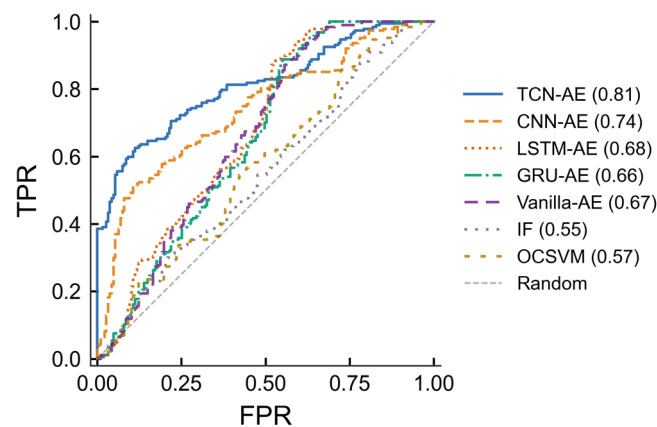


Figure 8. Receiver operating characteristic curves for all seven anomaly detection models. AUC values are annotated in the legend. TCN-AE attains the highest AUC (0.811).

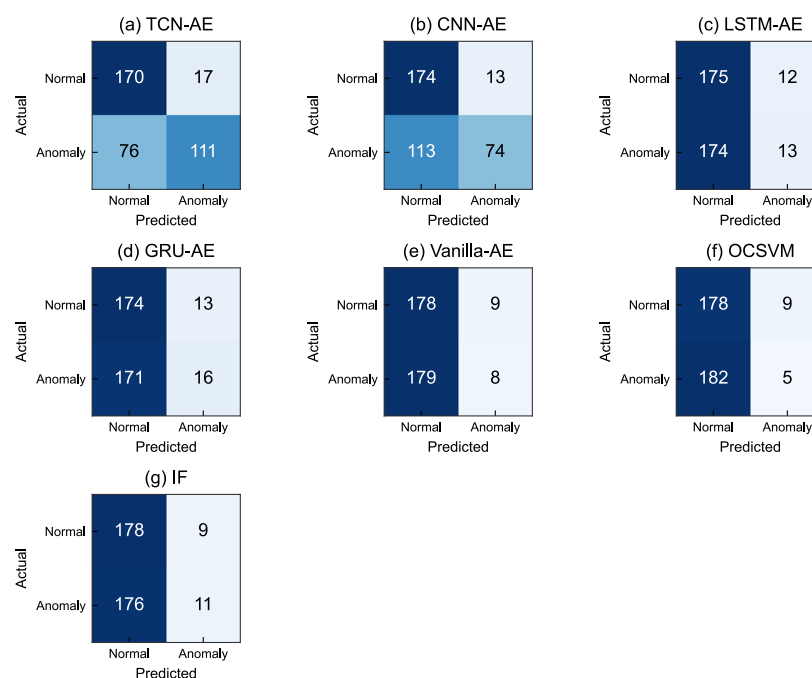


Figure 9. Confusion matrices for all seven models at the validation-calibrated P99 threshold. Each 2×2 matrix reports the counts of true negatives, false positives, false negatives, and true positives. TCN-AE achieves the highest true positive count with the lowest false positive rate.

The results reveal a two-tier structure correlating reconstruction precision with detection performance. Tier 1 comprises the convolutional architectures: TCN-AE (AUC 0.811) and CNN-AE (AUC 0.740). TCN-AE's low validation MSE ($\bar{e}_{\text{TCN}} = 2.7 \times 10^{-5}$) causes injected anomalies to produce a proportionally large MSE increase over the normal baseline, yielding well-separated score distributions. CNN-AE, with $\bar{e}_{\text{CNN}} = 1.6 \times 10^{-4}$, also achieves score separation (AUC 0.740) but trails TCN-AE. Tier 2 encompasses the recurrent and fully connected autoencoders together with the traditional methods, with AUCs from 0.55 to 0.68. Their higher reconstruction error ($\bar{e} \approx 2 \times 10^{-3}$ to 2×10^{-2}) stems from the limited ability of global hidden states to reproduce fine-grained local temporal patterns; consequently, the moderate anomaly amplitudes increase MSE by only ≈ 15 –25% over the normal baseline. Among the Tier 2 deep learning models, LSTM-AE achieves the highest AUC (0.680), followed by Vanilla-AE (0.665) and GRU-AE (0.659). Isolation Forest and One-Class SVM, operating on flattened 720-dimensional feature vectors, attain the lowest AUCs (0.550 and 0.569).

At the P99 threshold, Tier 2 models exhibit low F1 scores (0.050–0.148), reflecting the P99 threshold's design goal of controlling false positives rather than maximizing per-model F1. Their precision ranges from 0.357 to 0.552, with recall below 0.09 for all models, confirming that single-channel reconstruction errors from these architectures carry limited discriminative information at any fixed threshold.

Figure 10 illustrates the two-tier phenomenon. For TCN-AE, the mean anomaly MSE ($\bar{e}_{\text{anomaly}} = 6.88 \times 10^{-4}$) is approximately 17 $\times$ the mean normal MSE ($\bar{e}_{\text{normal}} = 4.13 \times 10^{-5}$), producing well-separated score distributions. In contrast, for LSTM-AE, the mean anomaly MSE ($\bar{e}_{\text{anomaly}} = 1.64 \times 10^{-2}$) is only 1.2 $\times$ the mean normal MSE ($\bar{e}_{\text{normal}} = 1.42 \times 10^{-2}$), resulting in nearly indistinguishable distributions.

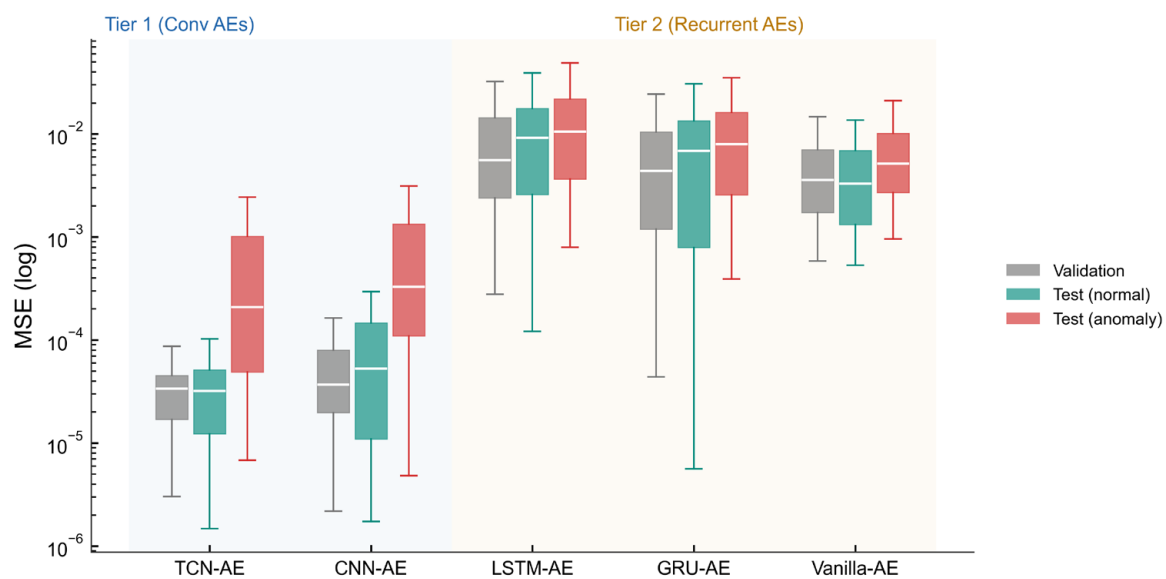


Figure 10. Distribution of per-sample reconstruction MSE for the five DL models across three conditions: validation (normal), test-normal, and test-anomaly. The logarithmic scale reveals the two-tier structure: Tier 1 models (TCN-AE, CNN-AE) show clear separation between normal and anomalous MSE distributions, while Tier 2 models (LSTM-AE, GRU-AE, Vanilla-AE) exhibit substantial overlap.

As shown in Figure 11, convolutional models outperform other model types. TCN-AE achieves the best performance across all anomaly types, with 100% recall on both spike and noise anomalies.

TCN-AE outperforms CNN-AE on both AUC ($\Delta\text{AUC} = +9.6\%$) and optimal F1 ($\Delta F_1 = +5.6\%$), with higher precision (0.679 vs. 0.622).

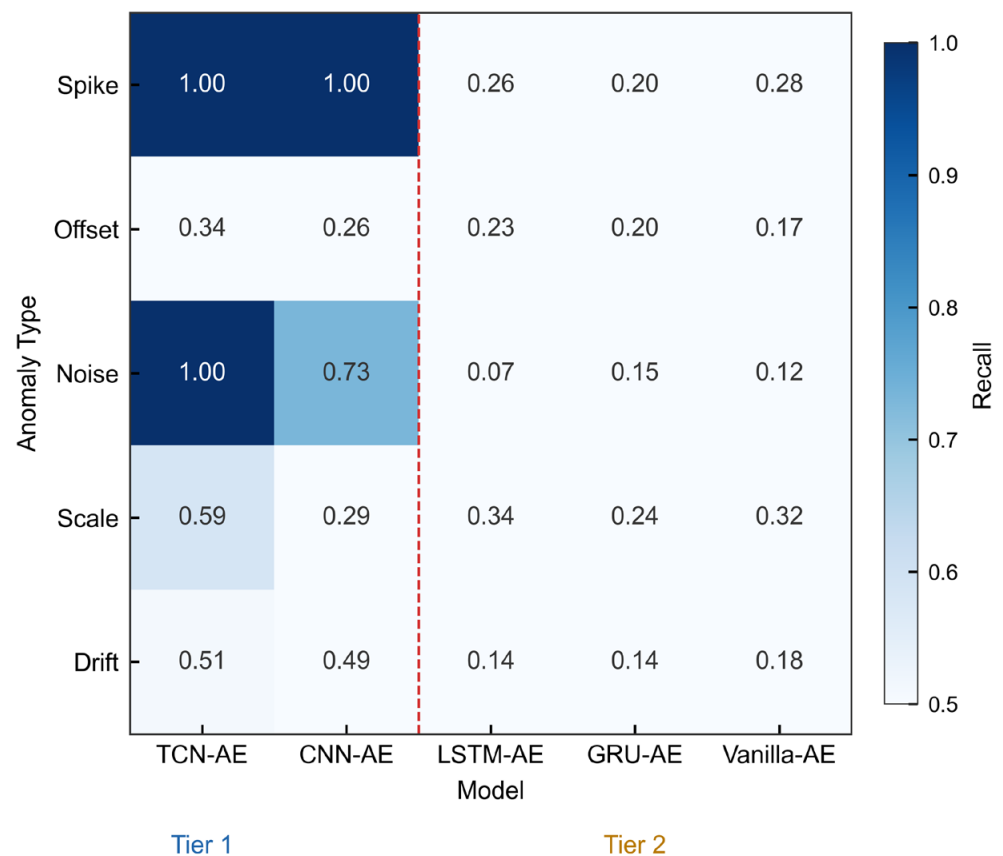


Figure 11. Recall of each model by anomaly type. TCN-AE and CNN-AE consistently outperform other models across all five anomaly types.

3.5. Visualization Analysis

3.5.1. Reconstruction Quality

As shown in Figure 11, scale and drift anomalies exhibit notably low recall under the P95 threshold. To investigate the cause, Figure 12 juxtaposes detected and undetected samples for each type. In the detected cases, the injected perturbation produces a clear discrepancy between the anomalous signal and the TCN-AE reconstruction, pushing the MSE above the detection threshold. In contrast, the undetected samples involve low-amplitude injections that the model reconstructs with near-normal fidelity, indicating that detection failures arise from the subtle magnitude of the injected anomalies rather than a lack of model sensitivity.

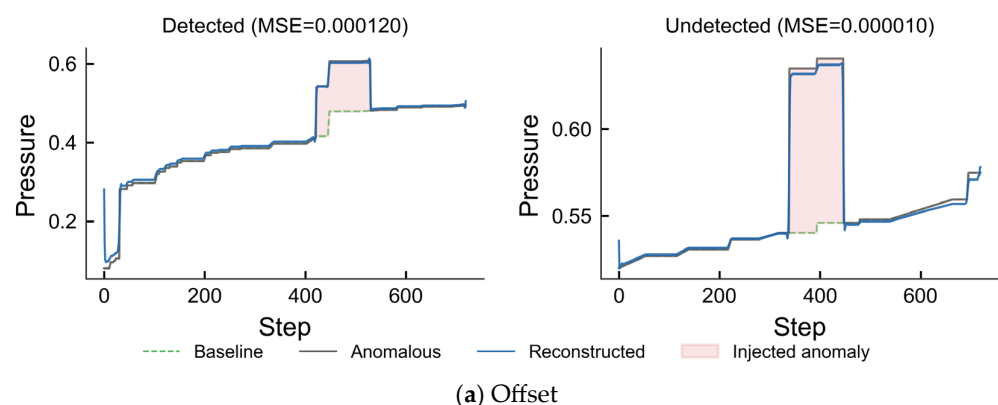


Figure 12. Cont.

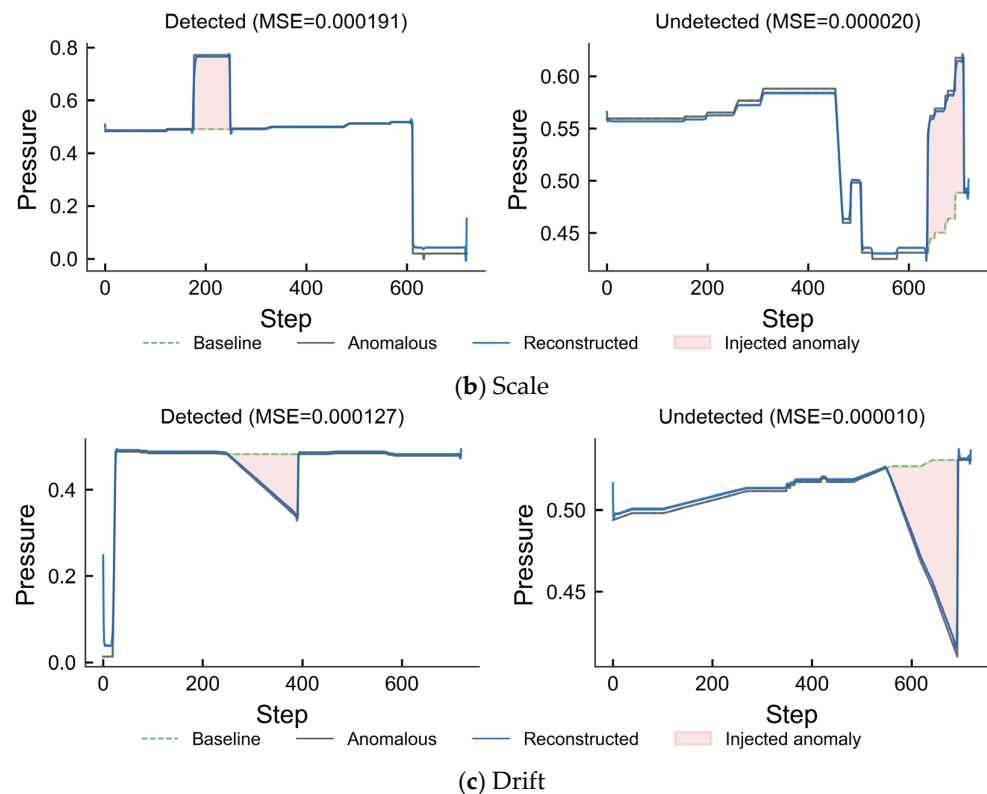


Figure 12. Detected (left) and undetected (right) reconstruction examples for three anomaly types: (a) offset, (b) scale, (c) drift. Green dashed lines denote the original baseline; gray lines the anomalous signal; blue lines the TCN-AE reconstruction. Red shading highlights the injected perturbation.

3.5.2. CUSUM vs. Static Threshold

Figure 13 illustrates the two strategies on the TCN-AE score sequence. Under the static threshold (top), only anomalous windows with MSE exceeding the fixed control limit are flagged, achieving high precision (0.862) but missing many anomalies whose individual MSE falls below the limit (recall 0.599). The CUSUM chart (bottom) accumulates positive deviations across consecutive anomalous windows, driving S_t past the decision interval h even for moderately elevated samples. This yields perfect recall (1.000) at the cost of reduced precision (0.763), resulting in a net F1 improvement from 0.707 to 0.866.

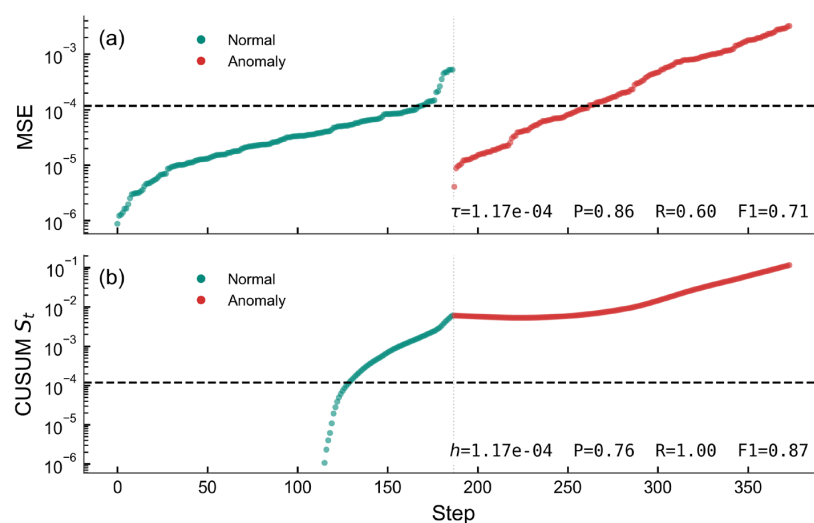


Figure 13. TCN-AE score sequences under static threshold (a) and CUSUM control chart (b) on the synthetic test set. Top: raw MSE per test window with the static control limit. Bottom: CUSUM

cumulative statistic S_t with a decision interval h . Each point is one test window; color indicates true label (green = normal, red = anomaly). Note the log scale on raw MSE.

3.5.3. Training Convergence

Figure 14 shows the training dynamics of all five deep learning models. All models are trained for a fixed 300 epochs, saving the checkpoint with the lowest validation loss. TCN-AE converges to the lowest validation loss ($\mathcal{L}_{\text{val}} = 2.7 \times 10^{-5}$), surpassing CNN-AE ($\mathcal{L}_{\text{val}} = 1.6 \times 10^{-4}$) by approximately an order of magnitude. LSTM-AE, GRU-AE, and Vanilla-AE plateau at 2–3 orders of magnitude higher validation loss.

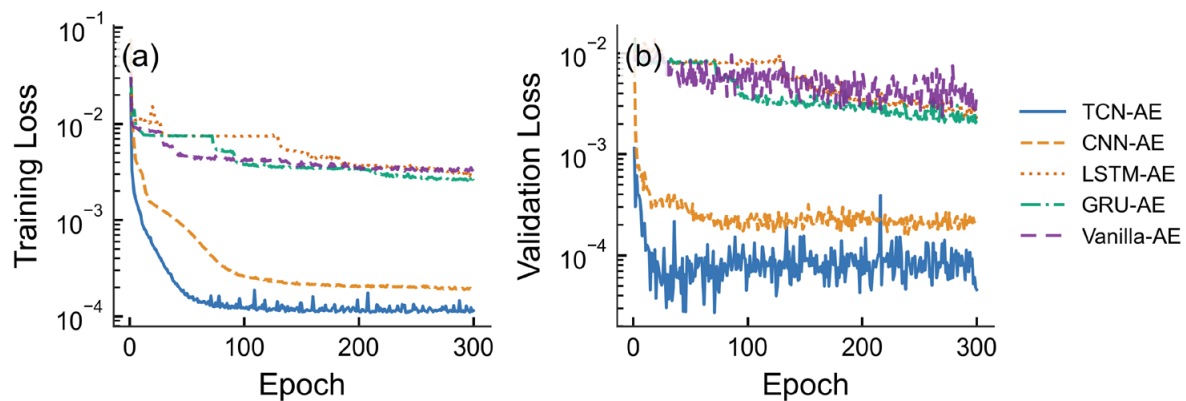


Figure 14. Training and validation loss (log scale) for all five DL models. (a) Training loss curves of all models over 300 epochs; (b) Validation loss curves of all models over 300 epochs. All models converge within 300 epochs. TCN-AE converges to the lowest validation loss, approximately an order of magnitude below CNN-AE.

3.6. Real Anomaly Evaluation

To assess the generalization capability of the proposed CUSUM framework beyond synthetic anomalies, we construct a separate test set from naturally occurring anomaly state records. The test set is constructed through a multi-stage filtering pipeline combining automated criteria and qualitative selection, to avoid the circular reasoning inherent in using model predictions to define ground truth.

3.6.1. Real Anomaly Dataset Construction

Records in the anomaly state mark periods where the hydraulic support is in an inactive or fault state. The test partition contains multiple contiguous blocks of anomaly state records. To isolate genuine fault signals, we apply a data-driven filtering pipeline. First, we detect interpolation artifacts by fitting a linear model to pressure values within each anomaly segment; sub-segments with a near-perfect linear fit ($R^2 > 0.97$) are excluded. Second, from each remaining segment, we extract non-overlapping windows of length $T = 720$ with stride $\Delta = 720$, discarding those with variance below 0.0002. This yields 204 candidate anomaly windows. Third, from the 204 candidates remaining after automated filtering, 26 windows exhibiting diverse fault morphologies (pressure drops, oscillatory failures, and abnormal fluctuations) are selected through visual inspection. An additional 12 windows are randomly sampled from the remaining candidates to reach a total of 38 anomaly windows, yielding a balanced test set of 76 samples (38 normal + 38 anomaly). An equal number of normal windows (38) is randomly sampled from non-overlapping segments of the test partition where the support state is not an anomaly. The automated filtering pipeline provides a reproducible, data-driven first stage with two objective criteria: $R^2 > 0.97$ for linear interpolation artifact removal, and window variance ≥ 0.0002 for near-constant signal exclusion.

The resulting balanced test set contains 76 samples (38 normal + 38 anomaly), constructed with seed = 42. All windows are normalized using the training-set MinMax parameters ($x_{\min} = 0.0$, $x_{\max} = 52.2$).

3.6.2. CUSUM Framework Results

We evaluate the CUSUM framework on this real fault test set using a 2×2 factorial design (TCN vs. CNN encoder $\times$ CUSUM vs. static threshold). Windows are arranged in online deployment order, reflecting realistic conditions where extended normal operation precedes fault onset. The quantitative results are summarized in Table 4.

Table 4. CUSUM Component Analysis on Real Fault Test Set (deploy order: normal $\rightarrow$ anomaly).

Encoder	Threshold	Precision	Recall	F1-Score	AUC-ROC
TCN-AE	CUSUM	0.826	1.000	0.905	0.586
TCN-AE	Static	0.556	0.132	0.213	0.586
CNN-AE	CUSUM	0.809	1.000	0.894	0.518
CNN-AE	Static	0.400	0.053	0.093	0.518

Figure 15 presents the ablation results as a grouped bar chart.

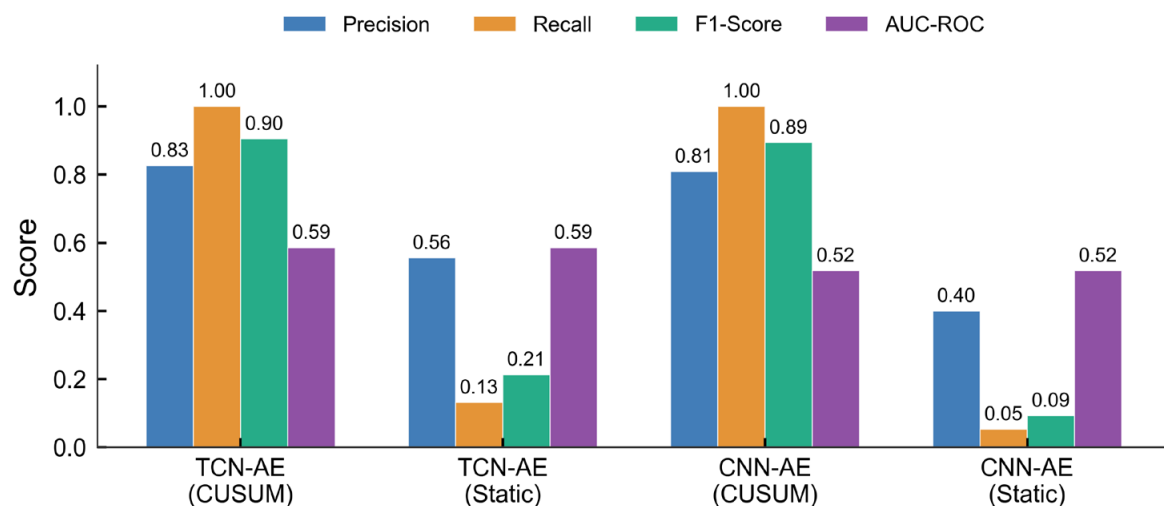


Figure 15. Component contribution analysis comparing CUSUM and static threshold strategies for TCN-AE and CNN-AE on the real fault test set under online deployment ordering. CUSUM achieves $F1 > 0.89$ for both models, improving over static by more than 0.69 absolute F1 points.

Under static thresholding, both models fail to detect most anomalies, with F1 scores of 0.09 to 0.21 and recall as low as 5.3% for CNN-AE. CUSUM's temporal accumulation achieves F1 scores of 0.905 (TCN-AE) and 0.894 (CNN-AE). This $\Delta F1 > 0.692$ improvement is even larger than the gains observed on synthetic anomalies ($\Delta F1 \approx 0.16\text{--}0.30$), demonstrating that CUSUM's value increases precisely when per-window discriminability is weakest. The CUSUM statistic accumulates sustained positive deviations above the in-control baseline across consecutive anomalous windows, driving the Cumulative Sum past the decision interval h despite individual windows having MSE values that barely exceed the normal baseline.

Figure 16 visualizes the TCN-AE score sequence under both strategies.

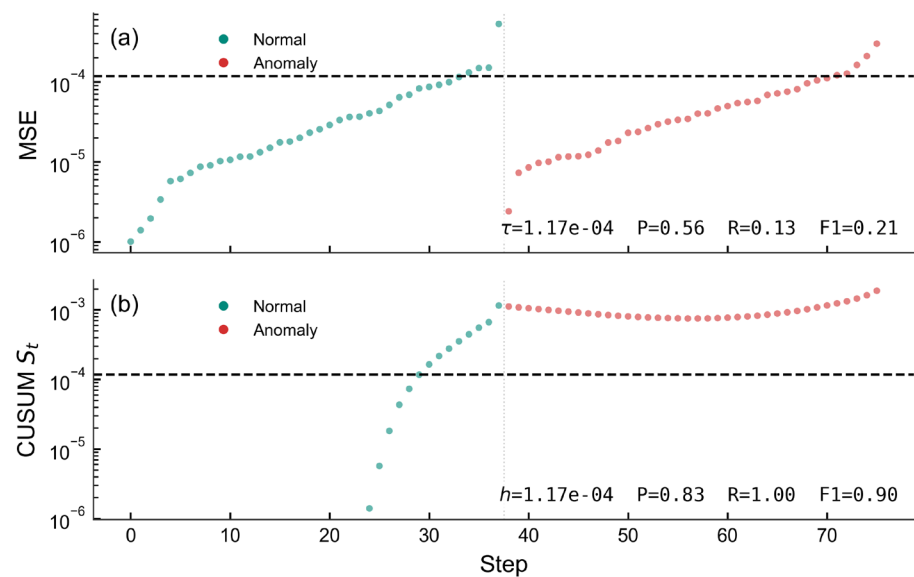


Figure 16. TCN-AE score sequences on the real fault test set under static thresholding (a) and CUSUM control chart (b). Under static thresholding, only 5 of 38 anomalies exceed the control limit (recall 0.132). The CUSUM chart accumulates sustained deviations, driving S_t past h for all anomalous windows while resetting during normal operation.

3.7. Case Study Analysis

To complement the quantitative evaluation, we present case studies on the manually selected real anomalies to illustrate detection mechanisms, model behavior differences, and failure modes at the individual-sample level.

3.7.1. Representative Real Fault Patterns

Figure 17 presents three representative real anomaly cases spanning a range of detection difficulty, along with TCN-AE reconstruction and per-step squared error.

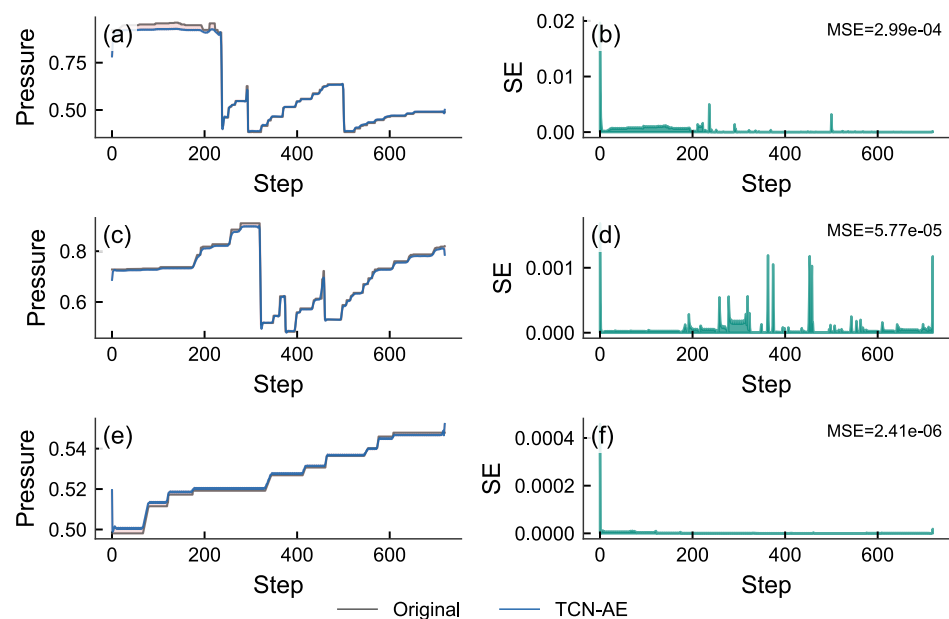


Figure 17. Representative real anomaly cases with TCN-AE reconstruction and per-step squared error. Left column: original signal (gray) vs. TCN-AE reconstruction (blue). Right column: per-step squared reconstruction error (green). Three cases (a–f) span easy-to-detect (top), moderately challenging (middle), and near-invisible (bottom) anomalies.

Case (a–b) shows a clearly abnormal pressure pattern with large-amplitude oscillations, producing a reconstruction error ($\text{MSE} = 2.99 \times 10^{-4}$) that visibly exceeds the normal baseline. Case (c–d) exhibits a moderate pressure anomaly where the reconstruction follows the general trend but deviates at the anomaly boundaries, yielding a moderate MSE. Case (e–f) presents a subtle deviation that is nearly indistinguishable from normal operation—the reconstruction tracks the signal closely, producing a low MSE comparable to normal windows.

3.7.2. Multi-Model Detection Comparison

Figure 18 compares the reconstruction error profiles from all five DL models on a single moderately challenging real anomaly sample.

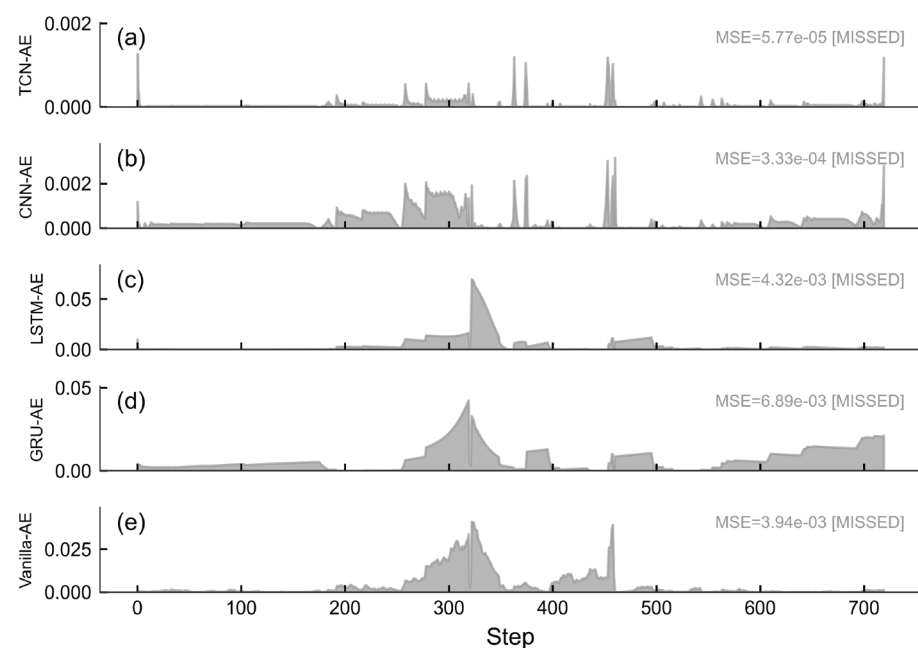


Figure 18. (a–e) Per-step squared reconstruction error from all five DL models on a moderately challenging real anomaly. Each subplot annotates the MSE value and detection status (DETECTED/MISSED) relative to the model’s P95 validation threshold. TCN-AE and CNN-AE detect this anomaly while the three recurrent models miss it, reflecting the Tier 1 vs. Tier 2 reconstruction precision gap observed on synthetic anomalies.

TCN-AE and CNN-AE detect this anomaly with reconstruction errors of 1.6×10^{-4} and 1.1×10^{-4} , respectively, both above their validation thresholds. The three recurrent models produce lower error magnitudes and miss the anomaly, consistent with the two-tier reconstruction precision phenomenon identified in the synthetic evaluation.

4. Discussion

4.1. Key Findings

Reconstruction precision is the primary determinant of detection AUC: TCN-AE and CNN-AE produce well-separated score distributions, whereas LSTM-AE, GRU-AE, and Vanilla-AE cannot reliably discriminate moderate-amplitude anomalies. TCN-AE achieves the highest AUC (0.811) versus CNN-AE (0.740) and all other baselines.

On the real fault test set, static per-window thresholding fails ($F1 = 0.213$), and the per-window AUC is near chance (0.586), confirming that individual reconstruction errors carry insufficient discriminative information for genuine operational faults. The CUSUM-based $F1$ of 0.905 is therefore attributable to temporal score accumulation rather than model

discriminability. CUSUM's temporal accumulation achieves F1 of 0.905 (TCN-AE) and 0.894 (CNN-AE)—a gain that far exceeds the synthetic anomaly improvement (+0.159), demonstrating that temporal score accumulation is most valuable precisely when per-window discriminability is weakest.

Detection difficulty differs substantially between synthetic and real anomalies; evaluation protocols relying on model predictions to define ground truth risk inflating performance estimates.

4.2. Limitations and Future Work

Several limitations warrant acknowledgment. First, the real anomaly test set is relatively small (76 samples, 38 per class), constrained by limited clean normal data in the test partition; expanding evaluation requires labeled data from multiple mining faces and extended monitoring periods. Second, the evaluation uses single-channel pressure signals, whereas real hydraulic faults may involve multi-channel interactions; extending the framework to multivariate inputs is a natural next step. Third, CUSUM achieves $F1 > 0.89$ on the real test set at the cost of non-zero false positives, requiring integration with domain expertise to filter residual alarms in operational deployment. Fourth, CUSUM's effectiveness is sensitive to baseline statistics (μ_0, σ_0) derived from the validation set, which may differ from the real-fault MSE distribution; adaptive online recalibration of these parameters is a promising direction. Fifth, the CUSUM evaluation in Section 3.3.3 considers artificially segmented alternating streams; real field deployment with irregular fault–normal duty cycles may exhibit different detection latencies. Sixth, the per-window decision granularity of 720 min limits the minimum detection latency to 12 h per decision. For applications requiring finer temporal resolution, a shorter window length with a higher overlap rate may be considered, though this would reduce the per-window information content and potentially affect reconstruction quality.

4.3. Why CUSUM over EWMA: A Negative Result Analysis

During method development, we initially adopted an Exponentially Weighted Moving Average (EWMA) smoothing filter as the score post-processing module. The EWMA statistic is computed as $z_t = \lambda \cdot s(\mathbf{x}_t) + (1 - \lambda) \cdot z_{t-1}$ with $\lambda = 2/21$ (span = 20), initialized from the in-control mean μ_0 . The EWMA applies decaying weights exponentially to past observations, acting as a low-pass filter intended to suppress transient MSE spikes in normal data.

Table 5 compares all three threshold strategies—static, EWMA, and CUSUM—under both the randomly shuffled and online deployment orderings for TCN-AE on synthetic anomalies.

Table 5. Threshold Strategy Comparison for TCN-AE (Synthetic Anomalies).

Scenario	Method	Precision	Recall	F1
Shuffled	Static	0.862	0.599	0.707
Shuffled	EWMA	0.519	0.963	0.674
Shuffled	CUSUM	0.508	0.984	0.670
Deploy	Static	0.862	0.599	0.707
Deploy	EWMA	0.919	0.604	0.729
Deploy	CUSUM	0.763	1.000	0.866

As shown in Figure 19, EWMA fails in both scenarios for different reasons. Under shuffled ordering, the EWMA smoother is pulled upward by anomalous samples and decays slowly, elevating the smoothed score for subsequent normal windows, producing a high false-positive rate ($P = 0.519$). Under deployment ordering, EWMA's smoothing delay causes the statistic to respond slowly to the onset of anomalies, keeping recall low at 0.604.

CUSUM, by contrast, achieves the best F1 in the deployment scenario (0.866) through its asymmetric design: the reset mechanism prevents inertial false positives during normal operation, while cumulative accumulation enables rapid detection when anomalies persist.

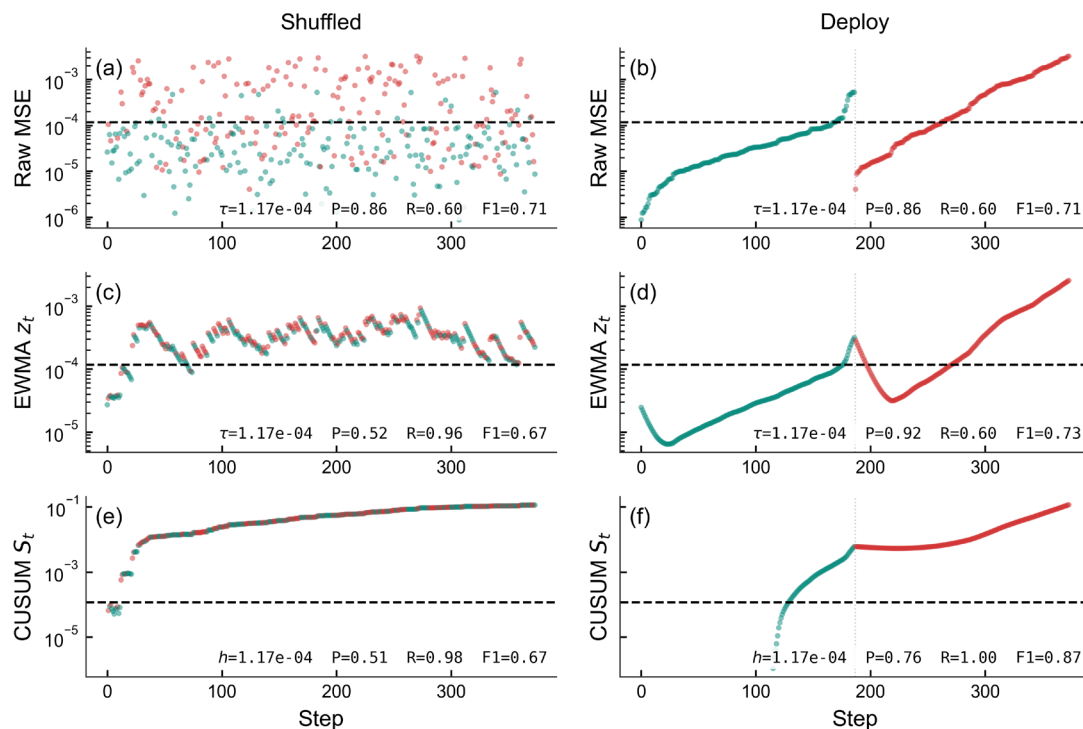


Figure 19. Comparison of three threshold strategies (static, EWMA, CUSUM) under two data orderings (shuffled, deploy) for TCN-AE on synthetic anomalies. (a) Raw MSE score sequence under shuffled ordering; (b) Raw MSE score sequence under deploy ordering; (c) EWMA smoothed score sequence under shuffled ordering; (d) EWMA smoothed score sequence under deploy ordering; (e) CUSUM statistic sequence under shuffled ordering; (f) CUSUM statistic sequence under deploy ordering. Each subplot shows the score sequence with true labels color-coded (green = normal, red = anomaly) and the respective decision threshold as a dashed line. Metrics are annotated in each panel.

Both EWMA and CUSUM depend on temporal structure, but CUSUM exploits it more effectively. This advantage is amplified on the real fault test set, where CUSUM's $\Delta F1 > 0.692$ over static thresholding far exceeds its synthetic anomaly gains.

4.4. Sensitivity of CUSUM Parameters

The CUSUM control chart depends on two parameters: the allowance $\delta = \gamma \cdot \sigma_0$ and the decision interval $h = \eta \cdot \sigma_0$ (where γ and η are dimensionless multipliers, as defined in Section 2.3.2). To assess sensitivity, we vary $\gamma \in \{0.25, 0.50, 0.75, 1.00, 1.50, 2.00\}$ and $\eta \in \{2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 10.0\}$ and evaluate F1 on the synthetic test set under deployment ordering.

The default configuration ($\gamma = 0.5$, $\eta = 5.0$) achieves $F1 = 0.866$ ($FAR = 0.310$). Within the recommended operating range $\gamma \in [0.25, 0.75]$, $\eta \in [4.0, 6.0]$, F1 ranges from 0.850 to 0.880, confirming acceptable stability. Tightening γ to 1.5 or above suppresses sensitivity (F1 decreases), whereas increasing η to 10.0 with $\gamma = 2.0$ yields the highest F1 of 0.942, but at an operational cost. The default ($\gamma = 0.5$, $\eta = 5.0$) provides a balanced trade-off.

The CUSUM formulation implicitly assumes that reconstruction errors are approximately i.i.d. under normal operation. Empirically, the first-order autocorrelation of the validation MSE sequence is 0.916, indicating strong temporal dependence. This violates the strict i.i.d. assumption and implies that the theoretical in-control average run length

(ARL_0) may not hold exactly. However, empirical results demonstrate that CUSUM remains effective despite this violation: at ($\gamma = 0.5$, $\eta = 5.0$), $F1 = 0.866$ is achieved on synthetic anomalies, and $F1 = 0.905$ on real faults. Future field deployments should monitor the serial correlation in the MSE stream and either adaptively recalibrate CUSUM parameters or consider residual whitening preprocessing to reduce autocorrelation.

5. Conclusions

This study presented TCN-AE, a temporal convolutional network autoencoder coupled with a CUSUM control chart, for online anomaly detection in hydraulic support pressure data. The principal findings are as follows.

- (1) The TCN encoder with dilated non-causal convolutions and residual connections achieves an AUC of 0.811 on synthetic anomalies, surpassing CNN-AE (0.740) and all recurrent (LSTM-AE 0.680, GRU-AE 0.659) and traditional baselines (Isolation Forest, One-Class SVM). Reconstruction precision (validation MSE $\sim 10^{-4}$) is the primary determinant of detection performance, producing a clear two-tier separation between convolutional and recurrent architectures.
- (2) The CUSUM dynamic threshold strategy accumulates sustained positive deviations across consecutive anomalous windows, achieving F1 improvements of +0.159 on synthetic anomalies and +0.692 on real faults for TCN-AE over static thresholding. On real faults, where per-window reconstruction scores carry near-random discriminability (AUC = 0.586), the gain is driven entirely by temporal accumulation. This demonstrates that CUSUM can provide operational detection capability even when per-window features lack discriminative power, though the resulting F1 reflects ordering advantage rather than model superiority.
- (3) A manually curated real fault test set reveals a substantial gap between synthetic and real anomaly detection difficulty, suggesting that evaluation protocols relying on model predictions to define ground truth may overestimate operational performance.

Author Contributions: Conceptualization, C.W. and W.X.; methodology, C.W.; software, W.X.; validation, C.W., W.X., and J.L.; formal analysis, C.W.; investigation, Z.H.; resources, J.L. and Y.Z.; data curation, X.Z.; writing—original draft preparation, C.W.; writing—review and editing, W.X. and J.L.; visualization, Y.Z.; supervision, J.L.; project administration, X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guizhou Provincial Department of Education's "Hundred Universities & Thousand Enterprises" Challenge Grant Program for Technological Breakthroughs, grant number [2024]013; the Guizhou Provincial Science and Technology Department's Support Program for Academic Development Assistance to Guizhou by High-Level Universities Outside the Province, grant number QiankeherencaiXKBF[2025]034; and the Guizhou Provincial Science and Technology Major Project, grant number Qiankehezhongda[2025]028.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to commercial confidentiality restrictions.

Acknowledgments: The authors thank the Guizhou Provincial Department of Education, the Guizhou Provincial Science and Technology Department, and the supporting programs for their financial assistance.

Conflicts of Interest: Authors Yu Zhao and Zhongguo He were employed by the company Guizhou Kailin Group Co. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Meng, Z.; Zeng, Q.; Gao, K.; Kong, S.; Liu, P.; Wan, L. Failure Analysis of Super-Large Mining Height Powered Support. *Eng. Fail. Anal.* **2018**, *92*, 378–391. [\[CrossRef\]](#)
- Zhao, X.; Li, F.; Liu, Y.; Fan, Y. Fatigue Behavior of a Box-Type Welded Structure of Hydraulic Support Used in Coal Mine. *Materials* **2015**, *8*, 6609–6622. [\[CrossRef\]](#) [\[PubMed\]](#)
- He, M.; Ma, X.; Yu, B. Analysis of Strata Behavior Process Characteristics of Gob-Side Entry Retaining with Roof Cutting and Pressure Releasing Based on Composite Roof Structure. *Shock Vib.* **2019**, *2019*, 2380342. [\[CrossRef\]](#)
- Li, W.; Ding, C. Design of Monitoring Platform of Hydraulic Support Operation Pressure Based on ARM and CAN Bus. In *Proceedings of the 2015 4th International Conference on Sensors, Measurement and Intelligent Materials*; Atlantis Press: Paris, France, 2016; pp. 501–505.
- Kim, D.; Heo, T.-Y. Anomaly Detection with Feature Extraction Based on Machine Learning Using Hydraulic System IoT Sensor Data. *Sensors* **2022**, *22*, 2479. [\[CrossRef\]](#) [\[PubMed\]](#)
- Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [\[CrossRef\]](#) [\[PubMed\]](#)
- Sakurada, M.; Yairi, T. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, MLSDA'14*; Association for Computing Machinery: New York, NY, USA, 2014; pp. 4–11.
- Prosvirin, A.E.; Ahmad, Z.; Kim, J.-M. Global and Local Feature Extraction Using a Convolutional Autoencoder and Neural Networks for Diagnosing Centrifugal Pump Mechanical Faults. *IEEE Access* **2021**, *9*, 65838–65854. [\[CrossRef\]](#)
- Tang, S.; Zhu, Y.; Yuan, S. Intelligent Fault Identification of Hydraulic Pump Using Deep Adaptive Normalized CNN and Synchrosqueezed Wavelet Transform. *Reliab. Eng. Syst. Saf.* **2022**, *224*, 108560. [\[CrossRef\]](#)
- Meng, L.; Zhao, M.; Cui, Z.; Zhang, X.; Zhong, S. Empirical Mode Reconstruction: Preserving Intrinsic Components in Data Augmentation for Intelligent Fault Diagnosis of Civil Aviation Hydraulic Pumps. *Comput. Ind.* **2022**, *134*, 103557. [\[CrossRef\]](#)
- Wu, X.; Zhang, Y.; Cheng, C.; Peng, Z. A Hybrid Classification Autoencoder for Semi-Supervised Fault Diagnosis in Rotating Machinery. *Mech. Syst. Signal Process.* **2021**, *149*, 107327. [\[CrossRef\]](#)
- Lindemann, B.; Fesenmayr, F.; Jazdi, N.; Weyrich, M. Anomaly Detection in Discrete Manufacturing Using Self-Learning Approaches. *Procedia CIRP* **2019**, *79*, 313–318. [\[CrossRef\]](#)
- Munir, M.; Siddiqui, S.A.; Dengel, A.; Ahmed, S. DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. *IEEE Access* **2019**, *7*, 1991–2005. [\[CrossRef\]](#)
- Chao, Q.; Shao, Y.; Liu, C.; Yang, X. Health Evaluation of Axial Piston Pumps Based on Density Weighted Support Vector Data Description. *Reliab. Eng. Syst. Saf.* **2023**, *237*, 109354. [\[CrossRef\]](#)
- Vapnik, V.N. An Overview of Statistical Learning Theory. *IEEE Trans. Neural Netw.* **1999**, *10*, 988–999. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zheng, X.; Wang, C.; Kong, C.; Liu, C.; Zhan, K.; Xu, R. Deep Learning Approach for Hydraulic Support Anomaly Detection: Utilizing Convolutional Autoencoders and Dynamic Time Warping Technology. *Rock Mech. Rock Eng.* **2024**, *in press*.
- Park, M.-H.; Chakraborty, S.; Vuong, Q.D.; Noh, D.-H.; Lee, J.-W.; Lee, J.-U.; Choi, J.-H.; Lee, W.-J. Anomaly Detection Based on Time Series Data of Hydraulic Accumulator. *Sensors* **2022**, *22*, 9428. [\[CrossRef\]](#) [\[PubMed\]](#)
- Neufeld, D.; Schmid, U. Anomaly Detection for Hydraulic Systems under Test. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*; IEEE: New York, NY, USA, 2021; pp. 1–8.
- Canizo, M.; Triguero, I.; Conde, A.; Onieva, E. Multi-Head CNN-RNN for Multi-Time Series Anomaly Detection: An Industrial Case Study. *Neurocomputing* **2019**, *363*, 246–260. [\[CrossRef\]](#)
- Wen, T.; Keyes, R. Time Series Anomaly Detection Using Convolutional Neural Networks and Transfer Learning. *arXiv* **2019**. [\[CrossRef\]](#)
- Chen, Z.; Chen, D.; Zhang, X.; Yuan, Z.; Cheng, X. Learning Graph Structures With Transformer for Multivariate Time-Series Anomaly Detection in IoT. *IEEE Internet Things J.* **2022**, *9*, 9179–9189. [\[CrossRef\]](#)
- Kim, K.; Jeong, J. Real-Time Monitoring for Hydraulic States Based on Convolutional Bidirectional LSTM with Attention Mechanism. *Sensors* **2020**, *20*, 7099. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zhu, L.; Laptev, N. Deep and Confident Prediction for Time Series at Uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*; IEEE: New York, NY, USA, 2017; pp. 103–110.
- Wang, S.; Xiang, J.; Zhong, Y.; Tang, H. A Data Indicator-Based Deep Belief Networks to Detect Multiple Faults in Axial Piston Pumps. *Mech. Syst. Signal Process.* **2018**, *112*, 154–170. [\[CrossRef\]](#)
- Shen, K.; Zhao, D. An EMD-LSTM Deep Learning Method for Aircraft Hydraulic System Fault Diagnosis under Different Environmental Noises. *Aerospace* **2023**, *10*, 55. [\[CrossRef\]](#)
- Huang, K.; Wu, S.; Li, F.; Yang, C.; Gui, W. Fault Diagnosis of Hydraulic Systems Based on Deep Learning Model With Multirate Data Samples. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6789–6801. [\[CrossRef\]](#) [\[PubMed\]](#)

27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2016; pp. 770–778.
28. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
29. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: New York, NY, USA, 2014; pp. 1724–1734.
30. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**. [[CrossRef](#)]
31. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**. [[CrossRef](#)]
32. Page, E.S. Continuous inspection schemes. *Biometrika* **1954**, *41*, 100–115. [[CrossRef](#)]
33. Gong, T.; Lee, J.; Cheng, X.; Xie, Y. Neural Network-Based CUSUM for Online Change-Point Detection. *arXiv* **2024**. [[CrossRef](#)]
34. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**. [[CrossRef](#)]
35. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
36. Liu, F.T.; Ting, K.M.; Zhou, Z.-H. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*; IEEE: New York, NY, USA, 2008; pp. 413–422.
37. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [[CrossRef](#)] [[PubMed](#)]

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