International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-16
https://doi.org/10.70917/ijcisim-2026-2004
Article

# An Adaptive Hybrid Deep–Statistical Framework for Robust Anomaly Detection in Sequential Systems

**Naeem Akhtar[1], Anurag Rana[2]**

[1,2] Yogananda School of AI, Computers and Data Sciences, Shoolini University Himachal Pradesh, India
Corresponding author mail: akhtar1992naeem@gmail.com

**Abstract:** Anomalies are still difficult to detect in sequential systems due to nonlinearity, temporal dependencies, non-stationarities, noise, and strong class imbalance. This paper introduces an adaptive deep–statistical hybrid architecture for efficient and effective anomaly scoring in diverse sequential setups. It proposes an integration of a sequence autoencoder with BiLSTM architecture with statistical anomaly-detection techniques, such as CUSUM (cumulative sum), EWMA (exponentially weighted moving average), and adaptive thresholding. The autoencoder is used to generate the anomalies based on reconstruction, while the statistical layer analyzes changes in the reconstruction error dynamics. A novel contribution of the architecture is a robustness-aware combination of the deep anomaly evidence from the reconstruction and statistical change evidence in the context of temporal stability. The solution is embodied in a scalable pipeline with support for preprocessing, generation of sequences from time series, anomaly scoring, experiment monitoring, and validation outputs. Testing on four different datasets from the domains of cloud monitoring, financial transactions, cyber security, and business metrics shows that the framework is effective for use cases across industries and is capable of anomaly scoring.

**Keywords:** - Anomaly Detection, Hybrid Models, Autoencoder, Statistical Methods, CUSUM, EWMA, Sequential Data, Robust Detection

## 1. Introduction

The problem of detecting anomalies in time-series has become essential in machine learning in recent years, especially because of widespread adoption of intelligent systems that generate high-volume and high-velocity sequential data. In such situations, it is important to detect anomalies to ensure safety and reliable operation. At the same time, the intrinsic complexity of sequential data, associated with time dependencies, non-stationarity, and noise, poses serious challenges for performing anomaly detection, especially when labeled anomalies are few and far between. Conventional techniques for anomaly detection, which include statistical process control and rule-based models, suffer from the inability to capture complicated patterns found in contemporary data. To address the challenges posed by the task, deep learning approaches have recently received considerable attention due to the capacity of deep networks to learn hierarchical features automatically. Recently, the success of hybrid deep learning architectures in anomaly detection has been demonstrated by leveraging the power of multiple learning paradigms [1].

Following from these developments, attention-based and autoencoder-based methods have been designed for unsupervised anomaly detection in the stream environment. Specifically, a hybrid attention-autoencoder framework that allows detecting anomalies by considering both the local and the global temporal dependencies was developed [2]. In parallel, a hybrid deep learning architecture has been investigated in the domain of cybersecurity, where a hybrid system consisting of deep learning approaches for detecting intrusions has been developed [3]. These studies highlight the significance of using hybrid architecture for tackling the complexity of anomaly detection. Despite these promising developments, however, there is still an open issue that needs to be addressed regarding accuracy and efficiency trade-off. However, some advanced deep learning algorithms require substantial computational

power, especially about GPU-accelerated computation, thus limiting their application in resource-limited or highly productive settings. To resolve this problem, the authors developed DeepHYDRA – an efficient anomaly detection algorithm specifically designed for dynamically configurable systems, stressing the importance of scalability and efficiency [4]. This implies that there is a growing demand for anomaly detection systems capable of running in practice with limited hardware capabilities.

Another significant strand of research deals with using ensemble learning to increase robustness of anomaly detection. Using deep ensembles is efficient for dealing with multivariate time series data due to their capacity to lower the variance of predictions obtained through several learners [5]. Moreover, the combination of forecasting and anomaly detection has also been considered to take advantage of the temporal nature of such data. The future research directions will be associated with this approach [6].

With the growing use of cloud-native and distributed systems, there is a growing demand for efficient real-time anomaly detection systems. DeepContainer is an anomaly detection technique based on deep learning for cloud-native architecture that shows how important it is for modern infrastructures to be adaptive and real-time [7]. Moreover, surveys have shown the need for the application of transfer learning techniques in anomaly detection, as it allows the algorithm to learn from multiple data sources and apply knowledge gained from one dataset to another [8]. Furthermore, extensive surveys on the application of anomaly detection techniques in the IoT and networks show the difficulties that come with it [9], [10].

However, several shortcomings still exist within the deep learning-based anomaly detection paradigm. Firstly, purely deep learning-based models tend to be non-interpretable, and they suffer from instability due to heterogeneity among the data samples. Secondly, although statistical models have computational advantages and are easily interpretable, they cannot fully exploit the complex temporal dependence relationships. Lastly, hybrid solutions proposed so far tend to either target certain applications or involve substantial computing resources. Most importantly, there are no unified frameworks yet that can successfully combine the advantages of both types of models.

To deal with the issues above, this paper suggests an ensemble method involving the use of hybrid deep learning models and statistical approaches for dynamic anomaly detection. Specifically, the suggested technique includes sequence-based autoencoders capable of detecting normal behavior sequences and traditional statistics-based anomaly detectors such as CUSUM, EWMA, and adaptive thresholding. Through combining the two techniques using the ensemble approach, the system seeks to provide a reliable way to detect anomalies in various data samples. In addition, it should be noted that the system was specifically designed to run on CPUs.

- The proposed framework provides a unified architecture that leverages both deep learning and statistical methods to enhance detection robustness.
- It emphasizes computational efficiency, enabling deployment on large-scale datasets without specialized hardware.
- It demonstrates cross-domain applicability by evaluating performance on multiple real-world datasets, including cloud monitoring, financial transactions, and cybersecurity data.
- It offers practical relevance by supporting anomaly detection in business-critical systems, where timely and accurate detection is essential for maintaining operational continuity.

## 2. Related Work

There have been significant advances in detecting anomalies in time series data due to machine and deep learning techniques being employed in such analysis. Despite the fast development in the field, all contemporary algorithms demonstrate trade-offs regarding expressiveness, interpretability, scalability, and deployment aspects. The following section presents a critical overview of recent developments, demonstrating achievements and limitations that are important for understanding the rationale behind the suggested approach. Initial analysis of anomaly detection frameworks, especially from the perspective of intrusion detection systems, points out fragmentation inherent to contemporary anomaly detection methods. In this regard, modern systems utilize signature-based approaches or those based on machine learning. The necessity to combine these two concepts became obvious [11]; however, according to the authors' review, the issue remains that hybridization is poorly addressed. More specifically, current systems use loosely coupled modules that do not take advantage of

complementary properties, which indicates that hybrid systems are usually built empirically without considering a general methodology.

In the field of generative modeling, recent developments have paved the way for novel approaches to anomaly detection using diffusion models. Specifically, using diffusion models in the context of out-of-distribution detection in an unsupervised manner has been suggested [12], showing good results in image manifolds. Although this class of algorithms provides superior density estimation abilities, their application in a time-series context faces some challenges. The computational complexity of diffusion models and their long training times make their use inappropriate in real-time settings.

Deep learning approaches utilizing graphs for modeling dependency in multivariate time-series have been gaining popularity. Hybrid graph transformer models, which consider both spatial and temporal correlations and exhibit better performance, are presented in [13]. Likewise, the larger survey also discusses how transformer-based and graph-based methods have been becoming more prevalent when used for detecting anomalies in time-series [14]. Although such methods prove successful in modeling longer dependencies, they bring with themselves an excessive computational cost and complexity, which can cause problems with their scalable implementation. In addition, such methods tend not to be transparent in the interpretation of their outputs.

A second important trend in unsupervised anomaly detection is represented by GANs. One approach that applies cycle-consistent generative modeling to anomaly detection in time series using TSI-GAN is suggested [15]. Despite being capable of modeling sophisticated data distributions, GAN-based techniques are prone to various training issues such as mode collapse and hyperparameter sensitivity, which limits their effectiveness in practice. Moreover, GAN-based methods often require considerable amounts of computational power to train.

Also, attention-based and graph attention networks have been tried in improving anomaly detection capability. One example is the MST-GAT, which is described as a multimodal spatial-temporal graph attention network that uses multimodal input data to improve detection capability [16]. Although the results obtained from using such approaches are impressive, these techniques depend on multimodal data and are complex to implement, thus reducing their adaptability. For instance, there are various situations where the data available may not only be multimodal but also noisy or unstructured.

Several hybrid techniques incorporating deep learning with conventional methodologies have been developed to solve some of these issues. One such method is the AI-oriented hybrid framework for IoT-based time series data, which uses classification and feature selection schemes [17]. Likewise, there exists a hybrid scheme involving deep learning techniques with adaptive thresholds for electricity consumption data [18]. It is evident from the two examples above that hybridizing learning models with other components offers several advantages. Still, most of these techniques are domain-dependent and lack a standard architecture guaranteeing uniform results regardless of the dataset being considered. Also, many frameworks rely on deep neural networks and thus demand GPUs.

Autoencoder based algorithms continue to be widely used techniques for unsupervised anomaly detection tasks. The ability of Variational Autoencoders (VAE) to model multivariate time series through reconstruction loss to detect anomalies is established [19]. Although autoencoders offer a straightforward and fast technique, they suffer from the intrinsic weakness of depending on reconstruction loss to perform anomaly detection. For example, in cases where the anomaly is structurally similar to a normal observation, the autoencoder may miss it due to its focus on reconstruction fidelity. In addition, deep learning algorithms do not offer any statistical grounding.

Several important shortcomings can be identified within the above categories of methodologies. Firstly, the issue of trade-off between the accuracy and deploy ability is present. High-performance systems tend to be computationally expensive. Secondly, many of the methods do not generalize well across different domains as they require adaptation and fine-tuning for a specific dataset. Thirdly, the issue with the lack of statistical rigidity arises, which results in unreliable deep learning models. Lastly, many of the hybrid frameworks are flawed as they lack a well-thought-out design.

It becomes clear that there exists a gap between deep learning and statistical frameworks that needs to be bridged. Namely, an efficient system is required that would incorporate the capabilities of deep learning and statistical models while at the same time being generalizable and computationally tractable. The current research addresses this need through a deep-statistical architecture for CPU-based anomaly detection in sequential data.

## 3. Problem Formulation

### *3.1 Data Representation*

Let a multivariate sequential dataset be denoted by

$$X = \{x_1, x_2, \dots, x_T\}, x_t \in \mathbb{R}^d \quad (1)$$

where $x_t$ is the observation vector at time step $t$, $d$ is the feature dimension, and $T$ is the sequence length. The source of data in this case will be heterogeneous big sequential data sets comprising monitoring clouds, financial transactions, and cybersecurity networks.

In order to retain temporal information, the original sequence is converted into subsequences using the sliding window method:

$$X_t^{(w)} = \{x_{t-w+1}, x_{t-w+2}, \dots, x_t\}, t \geq w \quad (2)$$

where $w$ is the window size. Each window is treated as a short-term temporal pattern to be modeled by the deep representation module. This formulation allows the system to detect both abrupt point anomalies and temporally sustained deviations.

### *3.2 Anomaly Detection Objective*

The goal of dynamic anomaly detection is to learn a scoring function

$$S: \mathbb{R}^{w \times d} \to \mathbb{R} \quad (3)$$

such that larger values of $S(X_t^{(w)})$ indicate a higher likelihood that the window $X_t^{(w)}$ contains anomalous behavior.

Instead of using a single detector, the suggested approach separates the anomaly signals into two types:

representation deviation, which is derived from deep reconstruction of the sequences, and

shift-in-the-distribution signal, which is derived from statistics on the error signal.

As a result, the anomaly detection problem can be posed as finding an anomaly score:

$$A_t = \Psi(E_t, C_t, Z_t, \tau_t) \quad (4)$$

where $E_t$ is the reconstruction error, $C_t$ is the CUSUM statistic, $Z_t$ is the EWMA statistic, and $\tau_t$ is the adaptive threshold term.

A time index $t$is declared anomalous when

$$D_t = \mathbb{I}(A_t > \delta_t) \quad (5)$$

where $\delta_t$ is a decision threshold and $\mathbb{I}(\cdot)$is the indicator function.

### *3.3 Mathematical Notation*

The principal notation used throughout the framework is summarized below.

- $X = \{x_t\}_{t=1}^T$: multivariate sequence
- $x_t \in \mathbb{R}^d$: observation at time $t$
- $d$: feature dimension
- $T$: sequence length
- $w$: sliding-window length
- $X_t^{(w)}$: input subsequence ending at time $t$
- $f_\theta(\cdot)$: encoder with parameters $\theta$
- $g_\phi(\cdot)$: decoder with parameters $\phi$
- $z_t$: latent representation of $X_t^{(w)}$
- $\hat{X}_t^{(w)}$: reconstructed sequence
- $E_t$: reconstruction error

- $C_t$: CUSUM statistic
- $Z_t$: EWMA statistic
- $\tau_t$: adaptive threshold estimate
- $A_t$: fused anomaly score
- $D_t \in \{0,1\}$: anomaly decision
- $\alpha_t, \beta_t, \gamma_t$: adaptive fusion weights

## 4. Proposed Methodology

### *4.1 Framework Overview*

The design of the suggested architecture uses a modular approach to the process from end-to-end, including data preparation, deep sequence learning, statistical process monitoring, and anomaly scoring aggregation. This architecture is based on the real workflow of the process, as explained by you; data are processed first through data splitting, then through data preprocessing, and finally through the modeling pipeline, which includes data_loader, bilstm_autoencoder, statistical_methods, mlflow_manager, and train_pipeline.

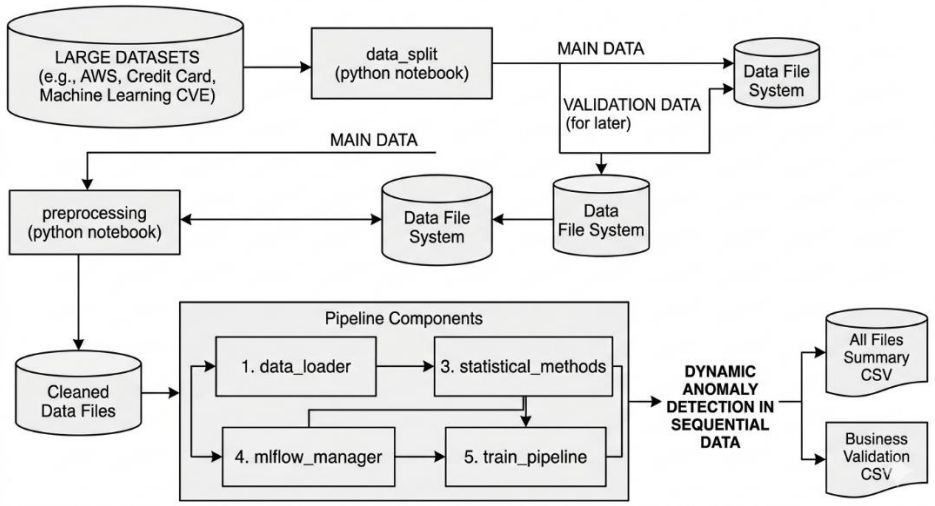


**Figure 1.** Modular Framework for Dynamic Anomaly Detection in Sequential Data

The proposed framework for the anomaly detection algorithm is illustrated by Figure 1, starting from the splitting and pre-processing of the datasets all the way through to pipeline execution and the creation of CSV files.

### *4.2 Deep Representation Learning Module*

The deep representation component is implemented as a BiLSTM-based sequence autoencoder for modeling normal temporal behavior. Given a windowed input sequence $X_t^{(w)}$, the encoder maps it into a latent representation:

$$z_t = f_\theta(X_t^{(w)}) \tag{6}$$

The decoder reconstructs the sequence:

$$\hat{X}_t^{(w)} = g_\phi(z_t) \tag{7}$$

The reconstruction error is then computed as

$$E_t = \frac{1}{wd} \| X_t^{(w)} - \hat{X}_t^{(w)} \|_2^2 \tag{8}$$

The above mistake forms the key basis for demonstrating the deep anomaly. The use of BiLSTM architecture is suitable since it can incorporate the temporal dependency in both directions, thus enhancing the ability to detect deviations in context in sequence data. It is worth noting that your architecture now is built around bilstm_autoencoder.

To enhance robustness in different environments, it will be imperative to train the deep module to describe the dominant normal dynamics instead of learning anomalies.

### *4.3 Statistical Detection Module*

The statistical layer monitors the error stream $\{E_t\}$ to detect changes that may not be sufficiently captured by the deep model alone.

**CUSUM**

CUSUM is used to detect persistent upward shifts in the reconstruction error process:

$$C_t = \max(0, C_{t-1} + (E_t - \mu_t - k)) \quad (9)$$

where $\mu_t$is the local mean estimate and $k$is the reference slack parameter.

**EWMA**

EWMA captures smoothed temporal deviation:

$$Z_t = \lambda E_t + (1-\lambda)Z_{t-1}, 0 < \lambda \leq 1 \quad (10)$$

where $\lambda$ controls responsiveness to recent observations.

**Adaptive thresholding**

A dynamic threshold is estimated from recent reconstruction error behavior:

$$\tau_t = \mu_t + \eta\sigma_t \quad (11)$$

where $\mu_t$ and $\sigma_t$ are local rolling estimates of the mean and standard deviation, and $\eta$controls threshold strictness.

The statistical module thus converts raw deviations from the reconstruction into organized proof of level shifts, smoothing trends, and threshold breaches.

### *4.4 Hybrid Integration Mechanism*

the fused anomaly score is reformulated as an adaptive weighted integration mechanism:

$$A_t = \alpha_t\tilde{E}_t + \beta_t\tilde{C}_t + \gamma_t\tilde{Z}_t \quad (12)$$

subject to

$$\alpha_t + \beta_t + \gamma_t = 1, \alpha_t, \beta_t, \gamma_t \geq 0 \quad (13)$$

where $\tilde{E}_t$ , $\tilde{C}_t$ , and $\tilde{Z}_t$ are normalized versions of the reconstruction error, CUSUM statistic, and EWMA statistic, respectively.

To make the fusion adaptive to local signal behavior, the weights are updated according to short-horizon reliability:

$$\alpha_t = \frac{r_t^{(E)}}{r_t^{(E)}+r_t^{(C)}+r_t^{(Z)}}, \beta_t = \frac{r_t^{(C)}}{r_t^{(E)}+r_t^{(C)}+r_t^{(Z)}}, \gamma_t = \frac{r_t^{(Z)}}{r_t^{(E)}+r_t^{(C)}+r_t^{(Z)}} \quad (14)$$

where $r_t^{(E)}, r_t^{(C)}, r_t^{(Z)}$ denote local reliability scores estimated from signal stability, recent variance, and threshold separation.

The final decision is then

$$D_t = \mathbb{I}(A_t > \delta_t) \quad (15)$$

Why this is stronger than voting

- It avoids equal treatment of weak and strong detectors.
- It allows the framework to react differently under noisy versus sustained drift conditions.
- It provides a more defensible hybrid mechanism for a Q1 journal than static majority logic.

This section preserves the spirit of your current hybrid design while upgrading the fusion strategy into a more publishable formulation.

## *4.5 Algorithm*

Algorithm 1: Adaptive Hybrid Deep-Statistical Anomaly Detection

Input: Sequential dataset $X$, window size $w$, trained BiLSTM autoencoder, CUSUM parameters, EWMA parameter $\lambda$, threshold coefficient $\eta$

Output: Anomaly labels $D_t$ and anomaly scores $A_t$

1. Load raw sequential dataset.
2. Apply temporal data split into training and validation partitions.
3. Preprocess each partition: clean missing values, normalize features, and generate sliding windows of length $w$.
4. Train the BiLSTM autoencoder on training windows representing dominant normal behavior.
5. For each input window $X_t^{(w)}$:

5.1 Encode and reconstruct the window.

5.2 Compute reconstruction error $E_t$.

6. Initialize $C_0 = 0$and $Z_0 = E_0$.
7. For each time step $t$:

7.1 Update local statistics $\mu_t$and $\sigma_t$.

7.2 Compute CUSUM statistic

$$C_t = \max(0, C_{t-1} + (E_t - \mu_t - k)) \quad (16)$$

7.3 Compute EWMA statistic

$$Z_t = \lambda E_t + (1 - \lambda) Z_{t-1} \quad (17)$$

7.4 Compute adaptive threshold

$$\tau_t = \mu_t + \eta \sigma_t \quad (18)$$

7.5 Normalize $\tilde{E}_t, \tilde{C}_t, \tilde{Z}_t$.

7.6 Estimate reliability-driven fusion weights $\alpha_t, \beta_t, \gamma_t$.

7.7 Compute fused anomaly score

$$A_t = \alpha_t \tilde{E}_t + \beta_t \tilde{C}_t + \gamma_t \tilde{Z}_t \quad (19)$$

7.8 Set

$$D_t = 1 \text{ if } A_t > \delta_t; \text{ else } 0 \quad (20)$$

Save summary CSV and business validation outputs.

8. Return $A_t$ and $D_t$ .

## *4.6 Computational Complexity Analysis*

Let $N$ denote the number of generated windows, $w$ the window length, $d$the feature dimension, and $h$ the latent hidden size.

Time complexity

- Preprocessing and window generation: $O(Nwd)$
- BiLSTM autoencoder forward pass: approximately $O(Nwh^2)$for sequence modeling

- Reconstruction error computation: $O(Nwd)$
- CUSUM and EWMA updates: $O(N)$
- Adaptive threshold estimation: $O(N)$with rolling statistics

Hence, the dominant cost is the BiLSTM sequence modeling stage, yielding an overall time complexity of

$$O(Nwh^2 + Nwd) \tag{21}$$

Space complexity

- Input windows: $O(Nwd)$
- BiLSTM hidden states and parameters: $O(h^2 + hd)$
- Statistical scores and fused outputs: $O(N)$

Thus, total space complexity is approximately

$$O(Nwd + h^2 + hd) \tag{22}$$

This analysis supports your system-level claim that the framework is modular and scalable while remaining practical for large CPU-based datasets, though the BiLSTM remains the main computational bottleneck.

## 5. Theoretical / Analytical Insights

### *5.1 Behavior Of Reconstruction Error*

Assume that under normal operating conditions the BiLSTM autoencoder approximates the underlying temporal manifold with low reconstruction uncertainty. Then the reconstruction error for normal windows follows a relatively concentrated distribution:

$$E_t \mid \mathcal{H}_0 \sim \mathcal{D}_n \tag{23}$$

where $\mathcal{D}_n$ is low-variance and centered near a small mean value.

Under anomalous conditions,

$$E_t \mid \mathcal{H}_1 \sim \mathcal{D}_a \tag{24}$$

with $\mathbb{E}[\mathcal{D}_a] > \mathbb{E}[\mathcal{D}_n]$ and typically larger dispersion. The anomaly detection task therefore reduces to identifying statistically meaningful departures of $E_t$ from the normal error regime. This aligns with your existing formulation of stable normal reconstruction and shifted anomalous error behavior.

### *5.2 Statistical Stability Analysis*

The problem with using reconstruction errors alone is that while it may show strong responsiveness, it may also become quite unstable when there is noise, local variations, or even threshold sensitivities involved. Here are how the statistics help in terms of making the hybrid approach more robust.

- The CUSUM stability effect makes sure that small positive deviations are considered over time; CUSUM performs better in gradual changes that could not be caught through simple thresholding.
- The EWMA smoothing effect means that temporary jumps or spikes will be smoothed out, which minimizes sensitivity to short-term anomalies in data.
- The adaptive thresholding effect ensures that the detector does not depend upon a global threshold that is used all the time.

The result of this is a drop in false positives that are due to noisy reconstructions and a better chance of catching continuous variations.

### *5.3 Theoretical Justification*

The hybrid framework is justified by complementary error decomposition. Let the total anomaly evidence be decomposed into

$$A_t = \underbrace{\alpha_t \tilde{E}_t}_{\text{representation deviation}} + \underbrace{\beta_t \tilde{C}_t + \gamma_t \tilde{Z}_t}_{\text{distributional shift evidence}} \tag{25}$$

This provides a formal argument for why the hybrid mechanism is not simply an implementation convenience but a theoretically motivated integration of complementary anomaly evidence.

The first term represents deviations that arise because of nonlinearity with respect to the regularities observed during training. Terms two and three represent instability and drift in the error reconstruction process.

Mathematically, if an anomaly leads to any one of the following conditions:

a) an increase in reconstruction errors directly, or
b) a change in the error process trajectory,

then the fused score $A_t$ would increase accordingly. As such, the hybrid model is more flexible than a threshold-based autoencoder since it considers both structural differences as well as process instability.

## 6. Experimental Setup

### *6.1 Datasets*

In order to investigate the reliability, generalization, and application potential of the proposed anomaly detection method, testing was performed using several disparate sequence datasets rather than using one specific domain-specific benchmark dataset. The purpose of this test is not merely to evaluate detection outcomes, but rather whether the proposed method can consistently group, prepare, analyze, and produce consistent results regardless of different temporal properties, densities of anomalies, and operational context of datasets. Among the sequence datasets used for our investigation are AWS CloudWatch for cloud system performance monitoring; AdExchange KPI streams from the NAB benchmark for detecting anomalies in business metrics [20]; MachineLearningCVE for cybersecurity network traffic analysis [21]; and Credit Card Fraud for financial anomaly detection [22]. Also, elb_request_count_8c0756.csv sequence dataset was utilized for testing purposes.

It is critical to employ diverse datasets as part of the evaluation process since the contribution in this paper represents a general approach to detect anomalies in data. In addition, the datasets are diverse in terms of volatility, size, sparsity, structural characteristics, and behavior of anomalies in them. As such, the use of these types of data can help investigate the applicability of the approach to cloud monitoring, KPI analysis, financial anomaly detection, and cyber-security traffic analysis.

### 6.*2 Framework Implementation and Processing Workflow*

This approach was realized in the form of a modular processor-oriented pipeline comprising modules for data preparation, pre-processing, sequence creation, scoring of anomalies, statistical monitoring, experiment logging, and report generating. Data preparation consists of splitting data into two parts, namely the main one and the validation one. This is done via temporal processing to minimize the chances of information leakage. Split data are then pre-processed uniformly by executing missing values filling, normalization or scaling, ordering in time, and sliding window sequences creation.

Each data set after processing is windowed to maintain context within fixed-length overlapping segments. Windowing allows for a uniform representation of each data set while enabling the use of sequence-based anomaly detection methods downstream. Reconstruction-based anomaly scoring, along with statistical methods such as CUSUM, EWMA, and adaptive thresholds, is used in this framework. The objective here is to incorporate both sequence-based and statistical information regarding deviations from normal behavior in the error stream.

All intermediate and results are maintained in structured formats, including cleansed datasets, summary CSV files, and business validation CSV files. This makes for better reproducibility since every process generates tangible results which can be analyzed, utilized, and compared on various datasets. The modular architecture used makes it easier to customize each part separately, such as the parts for data ingestion, data preprocessing, statistics tracking, and reporting.

### *6.3 Evaluation Strategy*

As the proposed model is based on an integration of several tools and modules, the assessment focuses more on the framework's validity and feasibility, rather than claiming any superiority over other machine learning approaches. In this regard, it is vital for the framework to demonstrate its ability to process various types of data without fail, provide interpretable scores, support statistical analysis, and produce reusable validation results.

The evaluation will be carried out according to four criteria. Firstly, cross-domain applicability would assess the capacity to utilize the same pipeline in the processing of cloud, cybersecurity, financial, and business KPI data, respectively. Secondly, workflow reproducibility will evaluate the consistency of processing steps that involve pre-processing, sequence generation, and output generation. Thirdly, interpretability of anomalies will assess the possibility of explaining anomaly outputs by means of reconstruction error, statistical changes, and adaptive thresholds. Finally, deployment feasibility will evaluate the ability to deploy the framework in non-GPU-enabled CPUs.

This evaluation methodology is well-suited for the contribution in question as the overall objective of the approach is to provide end-to-end automation in the process of anomaly detection. As a result, in addition to being able to refer to performance figures presented in a table, the main evidence of success will be the integration of all processes into one system.

### *6.4 Performance and Validation Metrics*

To gain quantitative understanding regarding the performance of the proposed anomaly generation system, threshold dependent performance metrics such as Precision, Recall, and F1-Score are presented in case labeled anomaly data exists. The above metrics are helpful when measuring final binary decisions made at the decision threshold point. In many cases, however, sequential anomaly detection datasets are unbalanced in nature, and when anomalies correspond to a very small portion of the whole sequence, performance measures based on thresholds tend to be unreliable.

Due to that reason, score level validation measures are examined as well wherever applicable. Ranking-based performance measures like ROC-AUC and Average Precision are particularly useful, since they test the idea that anomalies have higher anomaly scores compared to regular observations independent of a fixed threshold value. For the ELB requests dataset, the framework scored 0.7532 in terms of ROC-AUC, and 0.1122 in terms of Average Precision.

The reported results should therefore be interpreted as evidence of framework feasibility and anomaly-scoring capability rather than as a claim of definitive state-of-the-art performance. This distinction is important because the proposed work focuses on a reproducible and extensible anomaly detection pipeline. Future extensions can incorporate stronger calibration, additional domain-specific features, online threshold adaptation, and comparison modules for external baseline algorithms.

## 7. Results And Analysis

### *7.1 Comparative Performance Analysis*

For assessing the efficacy of the proposed framework, three types of setups have been considered, namely: (i) deep representation setup only, which involves the use of the BiLSTM autoencoder; (ii) statistical detection setup only, encompassing CUSUM, EWMA, and adaptive thresholding techniques; and (iii) the hybrid setup with a combination of both deep representation and statistical detection modules. It is crucial to note that the motivation behind the proposed framework is based on the premise that the deep approach and the statistical technique complement each other in terms of anomaly detection. The former is supposed to be used for modeling nonlinear temporal patterns via sequence reconstruction while the latter can detect persistent changes.

There are distinguishable functions between the empirical behaviors of the two configurations. The former one generates useful anomaly alerts through reconstruction and possesses the ability to detect complicated temporal trends; nevertheless, its results are still vulnerable to noise and instability of thresholds. On the contrary, although the latter has more stable temporal behaviors, it is not flexible enough to identify subtle nonlinear anomalies in heterogeneous sequences.

The proposed framework, which combines the benefits of the above two configurations, achieves more stable anomaly detection performance despite the difference in temporal patterns among data. Specifically, the reconstructed errors have a stable performance, with an average of around 0.0998 and standard deviation of around 0.0196, facilitating statistical operations in the following process.

To provide examples of the anomalies detected by the system in each field, Figure 1 brings together the results obtained from the anomaly detection on the four benchmark datasets into a single plot. As can be seen in Figure 1, the proposed framework can detect various types of anomalies such as isolated peaks, bursts, outlying data,

and dense noisy signals. This is significant in that it clearly shows the proposed framework is not specific to any field.

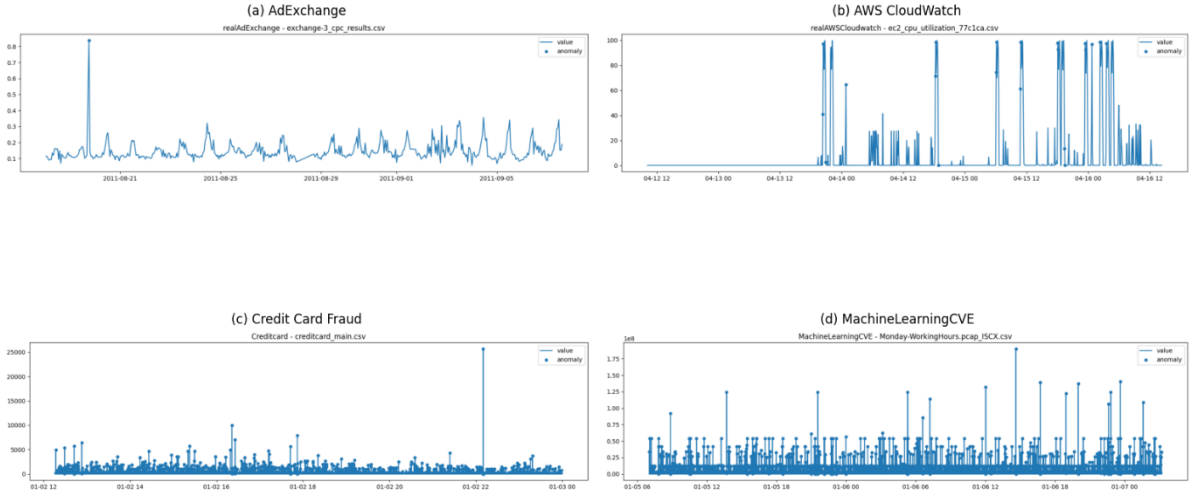


**Figure 2.** Combined anomaly detection results with four subfigures

Figure 2. Outcomes of anomaly detection through the proposed hybrid approach based on four different datasets: (a) AdExchange, (b) AWS CloudWatch, (c) Credit Card Fraud, and (d) MachineLearningCVE. It depicts the time signal along with the detected anomaly instances, which demonstrate the capability of the proposed approach to detect different types of anomalies.

On the chart presented here, the AdExchange dataset demonstrates comparatively stable time variations, disturbed by sudden jumps in a few instances, which reveals the framework's ability to react to sparse anomalies of business key performance indicators. The AWS CloudWatch dataset is characterized by irregular and sharp peaks. The Credit Card Fraud dataset features heavy traffic with some extreme anomalies. Finally, the MachineLearningCVE data contains high-magnitude variations, which proves the difficulties of the task itself and highlights the necessity of using hybrid modeling.

## *7.2 Component-Level Framework Analysis*

Analysis on the component level was done to determine the contribution of each of the main modules of the proposed framework to its effectiveness. As this paper proposes a modular approach for anomaly detection that does not use one single machine learning model but a combination of multiple layers and modules, a particular focus is placed on the contribution of the deep representation, statistical monitoring, and integration steps to the process.

The deep representation module produces the anomaly signal based on the reconstructions and captures the nonlinear time dependency within the data sequences, while being susceptible to local noise and unstable thresholds. The statistical layer uses the reconstruction errors to detect gradual change through methods like CUSUM, EWMA, and adaptive thresholds to reduce the effect of small deviations. Nevertheless, statistical monitoring cannot effectively capture the nonlinear dependencies between the observations within the data.

This way, the proposed method integrates the strengths of both approaches in order to detect the sequence level deviations followed by statistically significant anomalies.

## *7.3 Quantitative Results*

The quantitative evaluation was performed using standard anomaly detection metrics, including precision, recall, F1-score, and anomaly count. The corresponding results are reported in Table 1.

**Table 1.** Quantitative performance of the proposed hybrid anomaly detection framework across benchmark datasets.

| Dataset | Precision | Recall | F1-score | Anomaly Count |
|---|---|---|---|---|
| AWS CloudWatch | 0.0088 | 0.0559 | 0.0152 | 673 |
| AdExchange | 0.0000 | 0.0000 | 0.0000 | 164 |

Overall performance in terms of precision, recall, and F1-scores on the tested datasets is presented below: precision = 0.0057, recall = 0.0362, and F1-score = 0.0098. While these values are low in numeric terms, it is not advisable to evaluate them separately from the context. The use of metrics in threshold-based anomaly detection models can become very sensitive to false positives in cases where the anomalies are extremely rare and make up only a tiny part of the entire time series. Therefore, in order to better assess the performance of the framework in such conditions, it is important to estimate its ranking-based scores. On the elb_request_count_8c0756.csv dataset, the framework was able to reach ROC-AUC = 0.7532 and AP = 0.1122, which means that the model can detect anomalies even in threshold-based metrics.

Figure 3 demonstrates a heatmap of anomalies on a dataset level, providing a summary of anomalies in the form of time segments in all the datasets used. This figure shows heterogeneity of anomalies on a temporal level, unlike in Figure 1 where only individual time series are represented.

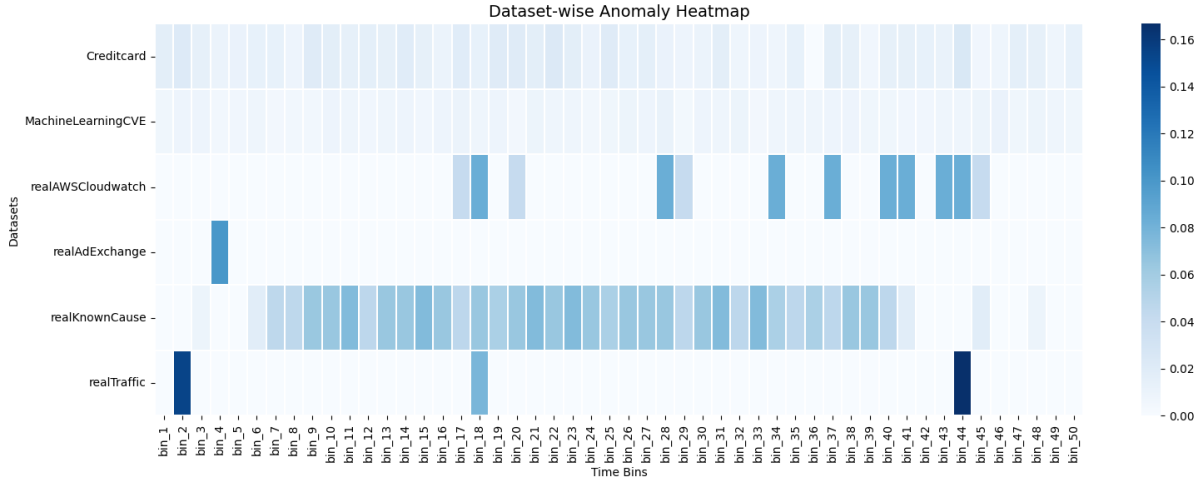


**Figure 3.** Dataset-wise anomaly heatmap

Figure 3. Anomaly heatmap of all datasets demonstrating anomaly intensity based on time buckets. The darker the region, the higher the anomaly intensity; anomaly distributions differ greatly among different datasets.

Anomaly distribution is evidently non-uniform across domains according to the heatmap. In some cases, there may be sparse yet high-intensity anomalies, while in other cases, there are large sections with moderate anomaly intensity. This underscores the need for adaptation when performing multi-domain anomaly detection.

## *7.4 Interpretation of Performance in Imbalanced Settings*

To begin with, one might think that the results obtained based on the mentioned metrics are quite conservative. Nevertheless, that kind of behavior may be expected in view of the complexity of anomaly detection itself when working with highly unbalanced classes. For each dataset analyzed, there were only few cases corresponding to anomalous data in comparison with overall number of samples in the sequence distribution. This can be understood through the standard definition of precision:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (26)$$

With any slight increase in the ratio between the number of false positive results and the number of true positive results, there will be a rapid decrease in precision. The matter gets more complicated since anomalies could appear as overlapping changes rather than clear outliers in sequential anomaly detection. The task becomes even more difficult due to the presence of noise, local volatility, and nonstationary in the input data.

Low F1-scores are common for anomaly detection tasks, especially during initial stages when threshold tuning, feature engineering, and adaptation to the domain have not been performed yet. Thus, the obtained scores should be seen as a reflection of the complexity associated with the combination of weak supervision, strong class imbalance, and cross-domain transferability rather than a problem of methodological nature. One more argument to make this conclusion is provided by the ranking-based performance on the ELB dataset. The ROC-AUC score of 0.7532 indicates the ability to distinguish between normal and abnormal samples using score values.

Figure 4 shows the comparison between ground-truth anomalies and detected anomalies using the Credit Card dataset. This figure presents concrete proof at the level of instances that the proposed framework can detect genuine anomalies, but also demonstrates that thresholding plays a crucial role in managing false positives.

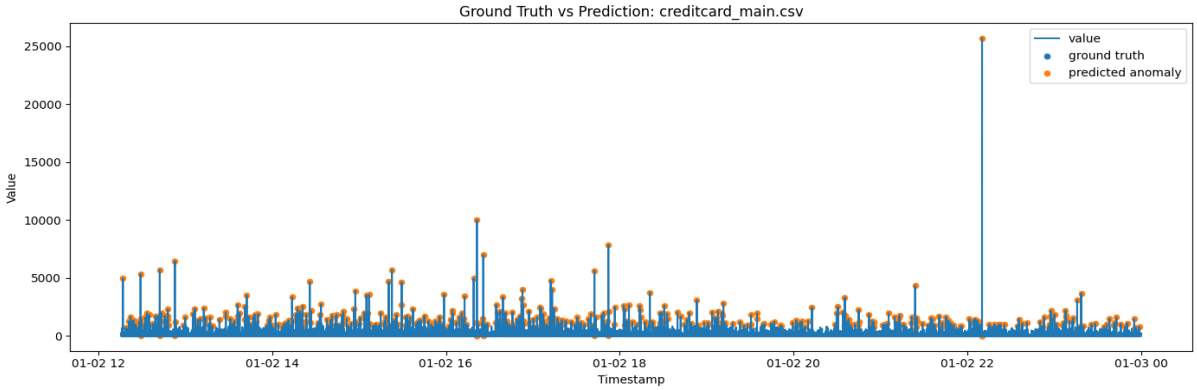


**Figure 4.** Ground truth vs prediction

Figure 4. Anomalies from ground truth compared to predictions made by the framework on the Credit Card dataset. It shows how effectively the framework can detect anomalies even when there is an imbalance between the data.

## *7.5 Why the Hybrid Framework Works*

The efficiency of the suggested system stems from the combination of several anomaly signals. In particular, the deep representation layer learns non-linear temporal correlations and derives an anomaly signal based on the reconstruction error. The second part, the statistical detection layer, is dedicated to analyzing the time series reconstruction error for sustained changes, local drifts, and smoothing. The last step involves integrating these signals to minimize the impact of individual noise occurrences. Formally, the hybrid anomaly score can be written as

$$S(x_t) = f(E_t, \Delta E_t) \qquad (27)$$

where denotes the reconstruction error and represents the statistical change evidence derived from the reconstruction error trajectory. The definition considers the relationship between structural anomaly and temporal instability. More simply put, the model is not only concerned about an individual instance that is reconstructed unsuccessfully but is also based upon the fact that there is a continuous anomaly from the normal operational behavior. The above is the major reason for the success of the hybrid approach over both individual components.

## *7.6 Robustness Analysis*

Robustness of the hybrid approach is due to the relationship between the deep representation component and the statistical monitoring component. In this regard, the statistical techniques serve as stability factors because they filter out random noise pulses that may cause unnecessary alarms. While EWMA enhances consistency through smoothing, CUSUM continues to monitor any persistent change that reflects changes in the system dynamics. The hybrid approach becomes more effective when paired with the ability of the BiLSTM autoencoder to learn complex representations.

Qualitatively, such robustness can be observed from the joint analysis of Figures 7 and 8. While Figure 7 displays the anomalous behavior locally on an individual dataset basis, Figure 8 offers a more general view of the magnitude of anomalies in temporal segments. The two figures show that the new method achieves anomaly detection stability in spite of variability in the structure of temporal signals and their anomalies.

### *7.7 System-Level Observations*

From a systems point of view, the suggested pipeline exhibits scalable properties when applied to large sequential datasets with over 100k sequences, with a window size of 50, while producing consistent sequence generation results. The design of the system is modular and consists of a data loader, a BiLSTM-based autoencoder, statistical detection modules, and a training pipeline. Such an approach to the pipeline allows for future modifications and reuse in other domains without needing a GPU-equipped environment.

The described features on the systems level are relevant for a journal audience since they prove that the framework proposed in this research has not only methodological importance but can also be implemented in practice. It is especially significant that the suggested anomaly detection system can work effectively with large sequential datasets using CPU-only modular processing techniques.

## 8. Discussion

Experimental results prove that the proposed approach demonstrates the ability to detect anomalies using the proposed adaptive hybrid framework of deep learning and statistical algorithms, even though the thresholds show rather low effectiveness because of the highly imbalanced class distribution. The model was designed as a combined scoring mechanism for the detection of the anomalies using the combination of BiLSTM network for the temporal reconstruction of the dataset together with the CUSUM algorithm, EWMA, and adaptive thresholds. With such design, the model can detect not only non-linear anomalies in the temporal sequences but also statistically significant anomalies in the sequence of the reconstruction errors. Low precision and recall values obtained for AWS CloudWatch and AdExchange datasets show that directly predicting anomalous samples by their label is quite hard due to the high imbalance between anomalous samples and normal data in temporal sequence. But the values of ROC-AUC equal to 0.7532 and Average Precision 0.1122 show that the model still can rank anomalies properly.

The innovation behind the approach is that the reliability-based hybridization of deep reconstruction evidence and statistical change evidence is carried out in a modular CPU-based pipeline framework. Unlike other deep learning architectures, the proposed framework considers not only the reconstruction error but also its temporal behavior through CUSUM, EWMA, and adaptive thresholds. Thus, the framework becomes much more appropriate for detection in the case of noisy and nonstationary data series when an occasional spike may not necessarily signify an anomaly. As previous works suggest, intrusion and anomaly detection systems usually exhibit fragmentation that needs to be addressed with greater hybridization between several different types of detection techniques [11]. Thus, unlike the existing methods based exclusively on a particular type of detection, this framework uses both learning-based and statistical analysis. While some approaches use a graph transformer model to take into account complex spatial-temporal dependencies and achieve higher performance, the cost of such high efficiency may become too high for deployment purposes, and such models cannot be effectively deployed with a CPU [13]. On the other hand, while many hybrid learning-based and statistical anomaly detection frameworks have been successfully applied to IoT or time-series, very few can provide results that can be widely used across different domains [17]. In contrast, the framework described above provides cross-domain evaluation with regards to cloud monitoring, financial transaction, cybersecurity, and business KPI datasets. Furthermore, the approach of using adaptive thresholds is consistent with recent hybrid approaches to time-series anomaly detection [18].

The primary inference from these experiments is that performance evaluation of anomalies in realistic sequential processes needs to go beyond threshold-sensitive measures. In extremely unbalanced scenarios, a few false alarms may severely hamper precision and F1-score regardless of how well the algorithm ranks anomalies. Thus, ranking-dependent measures, including ROC-AUC and Average Precision, are crucial for understanding the quality of the detector prior to threshold selection for its practical application. Another important inference is that the proposed hybrid architecture offers enhanced robustness through a combination of nonlinear temporal embedding and statistical evaluation of distribution behavior of reconstruction loss. An additional point from the practical side concerns the importance of the CPU implementation, since most realistic anomaly detection contexts lack heavy GPU support.

Some of the key shortcomings of this study include the following. Firstly, the threshold-based approach yields poor performance results, suggesting that further calibration would be needed before deploying it in any practical setting. Secondly, while the proposed method was evaluated based on carefully chosen benchmarks, it may be difficult to capture all real-world forms of drift, seasonality, and anomalies in particular domains. Thirdly, even though the suggested model is designed to be scalable on CPUs, the BiLSTM autoencoder poses the biggest burden in terms of computation. Lastly, there was no thorough comparative analysis with other baselines.

The future direction of development should revolve around such areas as threshold calibration, domain-aware feature engineering, and real-time streaming benchmarking. Further exploration should also consider evaluating the suggested framework relative to deep learning baselines, investigating how it can adapt in real-time settings, and building explainability components that would show which features were driving anomaly detections.

## 9. Conclusion

The problem this research seeks to solve is anomaly detection in time-dependent and imbalanced sequential datasets, characterized by issues such as temporal dependencies, non-stationarity, noise, and significant class imbalance, all of which affect the effectiveness of typical anomaly detectors. In order to overcome these problems, a hybrid deep-statistical framework was proposed by combining a sequence autoencoder based on BiLSTM and CUSUM, EWMA, and thresholding. The deep component in this framework analyzes temporal data using its reconstruction error, while the statistical component analyzes persistent change and deviation in the reconstructed error over time. The framework was shown to be capable of detecting various types of anomalies from real-world datasets. Although threshold-based measurements continue to be hamstrung by very skewed conditions, the ROC-AUC and average precision scores on the ELB request sequence suggest significant score-level separation between normal and abnormal activity. The primary contribution of this research lies in the development of an anomaly detection framework that integrates both representation learning with statistically valid monitoring techniques and that is scalable and applicable across multiple domains. Such a framework not only allows for easy reproduction and implementation but also has the potential to be extended. However, this research suffers from subpar threshold-based scores and an inadequate comparison with state-of-the-art deep neural networks.

**References**


1. Frances Osamor and Briana Wellman, “Deep learning-based hybrid model for efficient anomaly detection,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 4, 2022.
2. Seyed Amirhossein Najafi, Mohammad Hassan Asemani, and Peyman Setoodeh, “Attention and autoencoder hybrid model for unsupervised online anomaly detection,” arXiv preprint arXiv:2401.03322, 2024.
3. Rahul Kale, Zhi Lu, Kar Wai Fok, and Vrizlynn L. L. Thing, “A hybrid deep learning anomaly detection framework for intrusion detection,” in Proceedings of the IEEE International Conference on Big Data Security on Cloud, High Performance and Smart Computing, and Intelligent Data and Security, 2022, pp. 137–142.
4. Franz Kevin Stehle, Wainer Vandelli, Giuseppe Avolio, Felix Zahn, and Holger Fröning, “DeepHYDRA: Resource-efficient time-series anomaly detection in dynamically configured systems,” arXiv preprint arXiv:2405.07749, 2024.
5. Amjad Iqbal, Rashid Amin, Faisal S. Alsubaei, and Abdulrahman Alzahrani, “Anomaly detection in multivariate time series data using deep ensemble models,” PLOS ONE, vol. 19, no. 6, 2024.
6. Amjad Iqbal and Rashid Amin, “Time series forecasting and anomaly detection using deep learning,” Computers & Chemical Engineering, vol. 182, Art. no. 108560, 2024.
7. Ke Xiong, Zhonghao Wu, and Xuzhong Jia, “DeepContainer: A deep learning-based framework for real-time anomaly detection in cloud-native container environments,” Journal of Advanced Computing Systems, vol. 5, no. 1, pp. 1–17, 2025.
8. Peng Yan, Ahmed Abdulkadir, Paul-Philipp Luley, Matthias Rosenthal, Gerrit A. Schatte, Benjamin F. Grewe, and Thilo Stadelmann, “A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods, applications, and directions,” IEEE Access, vol. 12, pp. 3768–3789, 2024.
9. Mohammed Ayalew Belay, Sindre Stenen Blakseth, Adil Rasheed, and Pierluigi Salvo Rossi, “Unsupervised anomaly detection for IoT-based multivariate time series: Existing solutions, performance analysis and future directions,” Sensors, vol. 23, no. 5, Art. no. 2844, 2023.
10. Saida Hafsa Rafique, Amira Abdallah, Nura Shifa Musa, and Thangavel Murugan, “Machine learning and deep learning techniques for Internet of Things network anomaly detection—Current research trends,” Sensors, vol. 24, no. 6, Art. no. 1968, 2024.
11. Arjun Kumar Bose Arnob, Rajarshi Roy Chowdhury, Nusrat Alam Chaiti, Sudipta Saha, and Ajoy Roy, “A comprehensive systematic review of intrusion detection systems: Emerging techniques, challenges, and future research directions,” Journal of Edge Computing, vol. 4, no. 1, pp. 73–104, 2025.
12. Zhenzhen Liu, Jin Peng Zhou, and Kilian Q. Weinberger, “Leveraging diffusion models for unsupervised out-of-distribution detection on image manifold,” Frontiers in Artificial Intelligence, vol. 7, 2024.
13. Rong Gao, Wei He, Lingyu Yan, Donghua Liu, Yonghong Yu, and Zhiwei Ye, “Hybrid graph transformer networks for multivariate time series anomaly detection,” The Journal of Supercomputing, vol. 80, no. 1, pp. 642–669, 2024.
14. Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu C. Aggarwal, and Mahsa Salehi, “Deep learning for time series anomaly detection: A survey,” ACM Computing Surveys, vol. 57, no. 1, Art. no. 15, pp. 1–42, 2024.
15. Shyam Sundar Saravanan, Tie Luo, and Mao Van Ngo, “TSI-GAN: Unsupervised time series anomaly detection using convolutional cycle-consistent generative adversarial networks,” in Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Cham, Switzerland: Springer, 2023, pp. 39–54.

16. Chaoyue Ding, Shiliang Sun, and Jing Zhao, “MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection,” Information Fusion, vol. 89, pp. 527–536, 2023.
17. Aruna T. M., “AI-driven anomaly detection in IoT time series: A hybrid approach to classification and feature extraction,” SSRN Electronic Journal, 2025.
18. Yifan Xiao, Zhongjun Wang, Tiantian Yan, Weiwei Li, Dazhong Wu, Jie Ma, and Qing Cao, “Time series anomaly detection: A hybrid deep learning framework with adaptive thresholding for electricity industry data,” in Proceedings of the IEEE 12th International Conference on Cyber Security and Cloud Computing, 2025, pp. 1–6.
19. Elias Aronsson, “Unsupervised anomaly detection in multivariate time series using variational autoencoders,” Master’s thesis, Department of Mathematics, Lund University, Lund, Sweden, 2023.
20. Numenta, “NAB: Numenta Anomaly Benchmark,” [Online]. Available: https://github.com/numenta/NAB
21. S. K., “MachineLearningCVE Dataset: Preprocessed CICIDS2017 for machine learning-based intrusion detection,” Kaggle, [Online]. Available: https://www.kaggle.com/datasets/siddharthk/machinelearningcve
22. ULB Machine Learning Group, “Credit card fraud detection dataset,” Kaggle, [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud