HAD-CDA: A hybrid approach for anomaly detection in streaming data with concept drift adaptation

Sree Rama Chandra Murthy Akuri [a,1,*], Venkata Narayana Chejarla [b,2]

[a] Research Scholar, Dept. of CSE at JNTUK, Kakinada, A.P, India.
[b] Supervisor, Professor, Dept. of CSE at Lakireddy Bali Reddy College of Engineering, Mylavaram, A.P, India
[1] srcmurthy.akuri@gmail.com; [2] cvnreddy.chejarla@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received September 16, 2025
Revised December 17, 2025
Accepted December 18, 2025
Available online May 31, 2026

Keywords
Anomaly Detection
Streaming Data
Concept Drift
LSTM Autoencoder
Quant Tree-EWMA

ABSTRACT

Streaming anomaly detection is a difficult task because the data distribution is changing, and concept drift degrades the performance of traditional detection methods. To address the issue, this article proposes HAD-CDA (Hybrid Anomaly Detection with Concept Drift Adaptation), a combined system that detects local anomalies in time series using LSTM Autoencoders (LSTM-AE) and global anomalies in the distribution using Quant Tree-EWMA (QT-EWMA). The proposed framework accomplishes three objectives: (i) the use of the dynamic weighting mechanism, which automatically changes the contribution of each component (lambda between 0.2 and 0.8) according to their effectiveness, (ii) two concept drift detectors are proposed, i.e., the Kolmogorov-Smirnov and Page-Hinkley tests, to allow detecting concept drift; and (iii) the use of Elastic Weight Consolidation (EWC) to reduce catastrophic forgetting during update of the model. Experiments on four real-world streaming datasets, HTTP, SMTP, ForestCover, and Shuttle, indicate that HAD-CDA has AUCs of 0.95-0.97, an 8-9% improvement over state-of-the-art methods, and F1-scores of 0.81-0.95. The recall measure obtained by the LSTM-AE element is 0.8094, compared to 0.9399, and the specificity of QT-EWMA is very high at 0.9399. The framework is highly adaptable to different types of drifts, regaining performance levels around 9092 before drift, and 1525 windows would be needed by a baseline method, with a low processing latency of 12.430.9 ms per window. Having a per-sequence complexity of O(1), memory cost of O(n), low DIS (approximately 0.08), and stability indices of 0.02-0.04, the suggested HAD-CDA framework is an accurate, efficient, and robust solution to real-world streaming anomaly detection in changing data.

1. Introduction

The sudden increase in the size of the digital ecosystem has resulted in a significant explosion of data production, and the streaming data has become the main aspect of the contemporary information systems. Streaming data is generated constantly and at high speed and is commonly used in the financial market analysis [1], healthcare monitoring [2], Internet of Things (IoT) applications [3], and cybersecurity surveillance [4]. Streaming data, unlike the traditional batch data, comes in bit by bit, has severe temporal dependencies and is usually noisy and uncertain. Besides, the data distribution behind it can also change with time, a phenomenon called concept drift that significantly worsens the work of fixed learning models and traditional batch-based processing methods [5], [6]. Such features require low-latency, adaptive, and continuous learning and real-time decision making. The latest developments in the field of machine learning have enhanced the performance of streaming data analysis but critical

drawbacks still persist. Long Short-Term Memory (LSTM) and other Deep Learning architectures are useful for capturing complex temporal patterns and sequential dependencies [7]. But they have a high computational cost and can cause disastrous forgetting in response to changes in data distributions, making them unfeasible in real-time streaming applications [8]. Conversely, classic statistical methods such as Exponentially Weighted Moving Average (EWMA) are computationally intensive and respond rapidly to changes, yet are poor at modeling non-linear associations and cannot identify subtle or complex anomalies [9]. Consequently, current approaches tend to trade off among detection precision, versatility, and computational efficiency. Although much research has been conducted on streaming anomaly detection, several issues remain unresolved. The most commonly available frameworks are either good at defining temporal deviations or distributional outliers, but not both [10]. Besides that, most approaches use fixed thresholds or constant ensemble weights, which limit their ability to handle various forms of concept drift, such as sudden, gradual, and recurring drift [11]. Moreover, deep learning techniques tend to achieve higher detection accuracy but higher latency, whereas lightweight statistical techniques tend to achieve lower detection performance but higher latency [12].

The current paper provides a solution to these challenges, which consists of HAD-CDA (Hybrid Anomaly Detection with Concept Drift Adaptation), a new framework that combines the advantages of deep learning and statistical methods. The main contributions of this work are summarized as follows:

- We propose HAD-CDA, a unified framework that combines LSTM Autoencoders for local temporal anomaly detection with QT-EWMA for identifying global distributional changes in streaming data.
- The framework introduces a dynamic weighting mechanism along with dual drift detection using Kolmogorov–Smirnov and Page–Hinkley tests, enabling rapid adaptation to sudden, gradual, and recurring concept drift.
- We integrate Elastic Weight Consolidation (EWC) to reduce catastrophic forgetting during model updates, achieving stable performance under evolving data distributions with low latency suitable for real-time applications.

The rest of this paper is structured as follows: Section 2 reviews the related literature on streaming anomaly detection and concept drift adaptation. Section 3 offers a detailed description of the proposed HAD-CDA framework. In Section 4, the experimental results are provided and the performance of the proposed method discussed. Lastly, Section 5 summarizes the paper and outlines directions for future research.

2. Literature Review

2.1. Deep Learning Approaches for Anomaly Detection

The use of deep learning for anomaly detection has advanced considerably in recent years. He et al. [13] introduced LSTM-based encoder–decoder models for multi-sensor anomaly detection, achieving 85–90% accuracy on industrial IoT datasets. Their work demonstrated the effectiveness of learning temporal patterns, but the model's performance decreased when the data distribution changed over time. Malhotra et al. [14] improved LSTM models by incorporating attention mechanisms, thereby enhancing the capture of long-term dependencies and achieving 92% precision on time-series benchmarks. However, this approach required extensive hyperparameter tuning and showed reduced performance under concept drift.

More recently, transformer-based models have achieved 94–96% F1-scores on static datasets [15], [16], but their $O(n^2)$ computational complexity and latency exceeding 100ms limit their suitability for real-time streaming applications. Graph neural network–based methods have also been explored for anomaly detection in network-structured data [17], but their applicability is often limited to specific domains and does not generalize well across different data types.

2.2. Hybrid and Ensemble Methods

Hybrid approaches have shown strong potential in improving anomaly detection by combining the strengths of different techniques. Jeffrey et al. [12] demonstrated the effectiveness of hybrid models in

cyber–physical systems by integrating machine learning, threshold-based, and signature-based methods. They reported 22-45% AUC superiority with their framework compared to single-method baselines, underscoring the complementary value of different detection strategies.

Ginni and Chakravarthy [18] proposed a combined framework that uses supervised and unsupervised learning, along with advanced feature engineering, achieving 97 percent accuracy on benchmark datasets. Although effective, their method relies on labeled data for supervision, which limits its use in fully unsupervised and continuous-streaming applications.

Streaming data analysis has also been extensively implemented by ensemble learning methods. Gomes et al.[10] proposed Adaptive Random Forests to evolve data streams, which includes online bagging and drift detection technologies. Their algorithm has shown consistent performance across various drift conditions, although it has quite high processing latency above 50 ms, which is limiting in real-time use.

2.3. Concept Drift Adaptation Techniques

One of the significant problems with streaming data analytics is concept drift because data distributions keep shifting. Ferreira et al. [19] introduced a detailed drift detection model that classifies drift as sudden, gradual, incremental, and recurring and comments on the appropriate ways to adapt to it. The taxonomy is a useful guide for the design of adaptive detection mechanisms and for the adaptive weighting strategy to be used in this work. Ding et al.[8] suggested an ensemble-based method to be used in non-stationary settings, which resulted in a 30 percent decrease in post-drift recovery time. Their findings reveal the success of using multiple learners; nevertheless, the method is not helpful in dynamically combining elements with changing data properties.

Zhang et al. [20] investigated dynamic ensemble learning with adjusting component weight and obtained good results on synthetic drift datasets. Although efficient, their approach relies on explicit drift-detection indicators, whereas the suggested framework enables more flexible, implicit adaptation to changing data patterns. Table 1 compares recent anomaly detection methods in terms of technique, dataset, performance, and limitations. Despite their strong performance, most methods remain vulnerable to concept drift, computational complexity, or limited adaptability, motivating the proposed HAD-CDA framework.

Table 1. Comparative Analysis of Recent Anomaly Detection Methods.

Method	Technique	Dataset	Performance	Key Limitation
LSTM-ED [18]	LSTM Encoder-Decoder	Industrial IoT	85-90% Accuracy	Concept drift vulnerability
Attention-LSTM [21]	LSTM + Attention	Time-series	92% Precision	High parameter sensitivity
Transformer-AD [22]	Self-attention	Static benchmarks	94-96% F1	$O(n^2)$ complexity
GNN-AD [14]	Graph Neural Networks	Network data	91% AUC	Domain-specific
Hybrid-CPS [15]	ML + Threshold + Signature	CPS	22-45% AUC gain	Multiple model overhead
Ensemble-NS [20]	Ensemble + Drift Detection	Synthetic	30%faster recovery	Explicit drift signals
HAD-MDGAT [23]	GAN + GAT	SMD	93% F1	Training instability

3. Method

Let's define the streaming data as a sequence of observations. $X = \{x_1, x_2, \ldots, x_t\}$arriving over time (Fig. 1). We partition the stream into overlapping windows. $W_i = \{x_i, x_{i+1}, \ldots, x_{i+n-1}\}$ of size n, where each window represents a local temporal context. The goal is to compute an anomaly score $S(W_i)$ that captures both [24]: **1) Local anomalies,** which get deviations from expected temporal patterns within the window**;** and **2) Global anomalies**, which capture deviations from the overall data distribution.

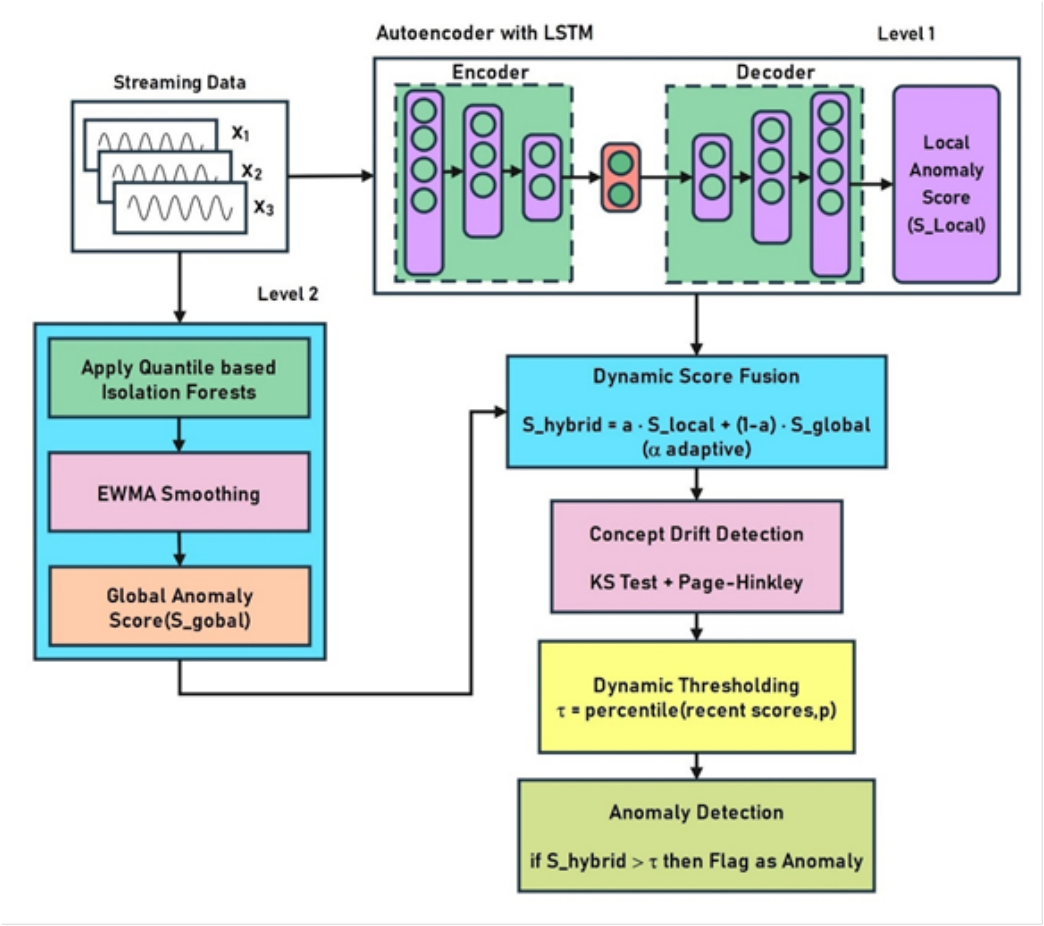

Fig. 1. Two-Level Hybrid Anomaly Detection Architecture for Streaming Data.

3.1. LSTM Autoencoder for Local Pattern Detection

LSTM AE learns a mapping function $f_\theta: R^{n\times d} \rightarrow R^{n\times d}$ (d represents the dimensionality of each data point in the time series, k represents the dimensionality of the latent space in LSTM AE parameterized by reconstructing the input window W_i:

$$\widehat{W}_i = f_\theta(W_i) \tag{1}$$

The local anomaly score is defined as the reconstruction error:

$$S_{local}(W_i) = \left\|W_i - \widehat{W}_i\right\|^2 \tag{2}$$

This component excels at capturing temporal dependencies but has limitations: It primarily models local patterns and may miss global distributional shifts, and it assumes the training data is representative, which may not hold under concept drift

3.2. QuantTree for Global Outlier Detection

QT-EWMA operates by transforming the window into a feature space using an Isolation Forest, computing an isolation-based score $Q(W_i)$ and applying EWMA smoothing [13]:

$$S_{global}(W_i) = \lambda Q(W_i) + (1-\lambda)\, S_{global}(W_{i-1}) \tag{3}$$

where λ is the smoothing factor. This component excels at detecting distributional shifts but may miss local temporal anomalies that don't affect the global distribution and does not explicitly model sequential dependencies. LSTM AE operates in the temporal domain, minimizing reconstruction error in the original feature space [25]

$$min_\theta \, \mathrm{E}_{W \sim D_{local}} [\|W - f_\theta(W)\|^2] \quad (4)$$

where D_{local} is the local temporal distribution.

QT-EWMA operates in the distributional domain, measuring isolation in a transformed feature space:

$$min_\theta \, \mathrm{E}_{W \sim D_{global}} \left[\mathrm{Iso}\left(g_\phi(W)\right)\right] \quad (5)$$

where $g_\phi$is the feature transformation and $Iso(\cdot)$ is the isolation measure. The point is that the two purposes are orthogonal: reducing reconstruction error does not always reduce isolation, thereby allowing the hybrid method to represent a larger range of anomalies [16].

3.3. Optimality of the Hybrid Approach

Information-theoretically, the goal of the hybrid approach is to maximize the mutual information between an anomaly's score and its actual state [26].

$$I(S(W); A) = H(A) - H(A \mid S(W)) \quad (6)$$

where A is the binary anomaly indicator and $H(\ \ )$ is entropy. The hybrid score $S_{hybrid}(W) = \alpha \cdot S_{local}(W) + (1 - \alpha) \cdot S_{global}(W)$ maximizes this mutual information because:

$$I(S_{hybrid}(W); A) \geq max(I(S_{local}(W); A), I(S_{global}(W); A)) \quad (8)$$

This inequality holds because the hybrid score contains information from both components, and the conditional entropy $H(A \mid S_{hybrid}(W))$ is minimized when both components contribute [23]. Let H_0 be the hypothesis that a window is normal, and H_1 be the hypothesis that it's anomalous. The optimal detector minimizes the Bayes risk:

$$R = P(H_0) \cdot P(false\ alarm) \cdot C_{FA} + P(H_1) \cdot P(miss) \cdot C_{miss} \quad (9)$$

where $C_{FA}$and C_{miss} are the costs of false alarms and misses, respectively. The likelihood ratio test is optimal:

$$\Lambda(W) = \frac{p(W|H_1)}{p(W|H_0)} \gtrless \gamma \quad (10)$$

where $\Lambda(W)$ is the likelihood ratio for the observation window W, $p(W \mid H_1)$ and $p(W \mid H_0)$ denote the conditional probability densities under the change and no-change hypotheses, respectively, and γ is the decision threshold determined by the prior probabilities and the relative costs of detection errors. A change is declared when $\Lambda(W) > \gamma$; otherwise, the observation is classified as belonging to the normal operating condition. In practice, the exact probability density functions are generally unavailable for complex multivariate data streams. Therefore, the proposed hybrid framework approximates the optimal LRT by integrating QT-EWMA for robust anomaly score estimation with CUSUM for sequential evidence accumulation, enabling reliable online change detection without explicit density estimation. where γ is a threshold. In practice, these densities are unknown; however, the hybrid approach can be shown to approximate this optimal test. $S_{local}(W)$ approximates $log_p(W \mid H_0)$ in the temporal domain. $S_{global}(W)$ approximates $log_p(W \mid H_0)$ in the distributional domain. The hybrid score combines these approximations:

$$S_{hybrid}(W) \approx \alpha \cdot logp_{temporal}(W \mid H_0) + (1 - \alpha) \cdot logp_{distributional}(W \mid H_0) \quad (11)$$

This combination provides a more comprehensive approximation to the true likelihood ratio than either component alone.

3.4. Dynamic Weighting Optimality

The dynamic weighting mechanism $\alpha_t = sigmoid(\beta \cdot (A_{local} - A_{global}))$ is optimal because it minimizes the expected error:

$$min_{\alpha} E\left[(\alpha \cdot S_{local} + (1-\alpha) \cdot S_{global} - S_{true})^2\right] \quad (12)$$

where S_{true} is the ideal anomaly score. Taking the derivative with respect to α and setting it to zero and solving for α:

$$\alpha = \frac{E[(S_{true} - S_{global}) \cdot (S_{local} - S_{global})]}{E[(S_{local} - S_{global})^2]} \quad (13)$$

It implies that the best weight is provided by the covariance between the true score and the component scores. The dynamic weighting mechanism approximates this relationship by varying based on recent performance, serving as a proxy for the actual covariance [27].

3.5. Handling Concept Drift

Concept drift occurs when the underlying data distribution changes over time: $D_t \neq D_{t+1}$. The hybrid approach addresses this through:

- **Online Learning with EWC:** The LSTM parameters θ are updated to minimize:

$$\Gamma(\theta) = \|W - f_{\theta}(W)\|^2 + \sum_i F_i \cdot (\theta_i - \theta_i^*)^2 \quad (14)$$

where F_i is the Fisher information matrix measuring parameter importance, and θ^* are the previous parameters. This prevents catastrophic forgetting while adapting to new patterns.

- **Dynamic Thresholding:** The threshold τ_t is updated as the ρ-percentile of recent scores:

$$\tau_t = percentile\left(\{S_{hybrid}(W_{t-k}), \ldots, S_{hybrid}(W_t)\}, \rho\right) \quad (15)$$

This adapts to changing score distributions caused by concept drift.

- **Dynamic Weighting**: The weight α_t adapts to changing conditions, placing greater emphasis on the component that performs better in the current context [28].

The hybrid framework integrates the LSTM AE and QT-EWMA models for anomaly detection in streaming data. Algorithm 1 presents the proposed Hybrid Anomaly Detection with Concept Drift Adaptation (HAD-CDA) framework. The algorithm integrates an LSTM Autoencoder for local anomaly detection and an Isolation Forest with QT-EWMA for global anomaly detection. The resulting anomaly scores are fused using adaptive weighting and evaluated through a dynamic thresholding mechanism.

To maintain robustness in evolving data streams, concept drift is continuously monitored using the Kolmogorov–Smirnov and Page–Hinkley tests. When drift is detected, the framework dynamically adjusts the fusion weights and incrementally updates both detection models using Elastic Weight Consolidation (EWC), enabling effective adaptation to changing data distributions while preserving previously learned knowledge.

The algorithm is designed for real-time processing, with time complexity of $O(1)$ per sequence and memory complexity bounded by $O(n)$, where n is the sliding window size. The hybrid framework is a good solution, combining local pattern detection using an LSTM AE with global outlier detection using QT-EWMA, delivering strong results for detecting anomalies in quantum streaming data. The EWMA smoothing integration greatly minimizes false positives, keeping anomaly scores constant over time, providing a trade-off between detection accuracy and computational efficiency in a large-scale streaming application [29].

Algorithm 1: Hybrid Anomaly Detection with Concept Drift Adaptation (HAD-CDA)

Input: *Data stream* $D = \{x_1, x_2, \ldots, x_T\}$; *Window size* n; *step size* s; *Initial weights* α , β; *EWMA smoothing factor* λ; *Threshold percentile* ρ; *Model update frequency* f; *EWC regularization strength* γ; *Drift detection window size* w_d ; *drift threshold* δ_d;

Output: Anomaly scores = $S = \left\{S_{hybrid}^{(1)}, \ldots, S_{hybrid}^{(T)}\right\}$
Anomaly flags $A = \{a_1, \ldots, a_T\}$
Drift indicators $D = \{d_1, \ldots, d_T\}$

Initialization: *Pre-train LSTM-AE on initial normal data batch*; *Initialize QT-EWMA with Isolation Forest on same batch*; *Initialize Fisher Information Matrix* $F \leftarrow 0$; *Initialize score history* $H \leftarrow \emptyset$; *Initialize threshold* $\tau \leftarrow 0$; *Initialize performance metrics* $P_{local} \leftarrow 0.5, P_{global} \leftarrow 0.5$; *Initialize drift history* $H_d \leftarrow \emptyset$;

For each time step $t = 1$ to T:

1. Window Extraction:
   If $t \geq n$, extract current window $W_t = \{x_t - n + 1, \ldots, x_t\}$
2. Local Anomaly Detection:
   Compute reconstruction: $\widehat{W}_t = LSTM - AE(W_t)$
   Calculate local score: $S_{local} = \left\|W_t - \widehat{W}_t\right\|^2$
3. Global Anomaly Detection:
   Compute isolation score: $Q_t = IsolationForest(W_t)$
   Apply EWMA smoothing:
   $$S_{global} = \begin{cases} Q_t & if\ t = n \\ \lambda Q_t + (1-\lambda) S_{global}^{t-1} & otherwise \end{cases}$$
4. Score Fusion:
   Compute hybrid score: $S_{hybrid} = \alpha S_{local} + (1-\alpha) S_{global}$
5. Dynamic Thresholding:
   Update score history: $H \leftarrow H \cup \{S_{hybrid}\}$
   Update threshold: $\tau \leftarrow percentile(H_{[-n:]}, p)$
6. Anomaly Detection:
   $$\text{Flag anomaly: } a_t = \begin{cases} 1 & if\ S_{hybrid} > \tau \\ 0 & otherwise \end{cases}$$
7. Concept Drift Detection:
   $$If\ t \geq n + w_d$$
   Extract recent scores: $S_{recent} = H_{[-w_d:]}$
   Extract historical scores: $S_{hist} = H_{[-(n+w_d):-n]}$
   Perform Kolmogorov-Smirnov test: $(KS, p_{val}) = KS(S_{recent}, S_{hist})$
   Perform Page-Hinkley test: $PH = PH(H, w_d)$
   $$\text{Detect drift: } d_t \begin{cases} 1 & if (p_{val} < 0.05 \wedge KS > \delta_d) \vee PH > \delta_d \\ 0 & otherwise \end{cases}$$
   if $d_t = 1$:
   Trigger adaptation: $\alpha \leftarrow 0.5$, $\tau \leftarrow percentile(H_{[-n:]}, p)$
   Record drift: $H_d \leftarrow H_d \cup \{t\}$
8. Dynamic Weight Adjustment:
   if $d_t = 1$:
   $$\alpha \leftarrow \sigma(\beta(P_{local}^{recent} - P_{global}^{recent}))$$
   else:
   $$\alpha \leftarrow \sigma(\beta(P_{local} - P_{global}))$$
9. Model Update:
   $if\ t\ mod\ f = 0$ or $d_t = 1$
   Update LSTM-AE with EWC:
   $$L = \|W_t - LSTM - AE(W_t)\|^2 + \gamma \sum_i F_i(\theta_i - \theta_i^*)^2$$
   Update Fisher Matrix: $F \leftarrow Fisher(LSTM - AE, W_t)$
   Update Isolation Forest: $QT - EWMA\ - update(W_t)$

3.6. Evaluation Metrics

The model performance was tested on a complete set of metrics that reflect detection accuracy and false positive control. Let TP denote true positives, TN true negatives, FP false positives, and FN false negatives.

a. Area Under ROC Curve (AUC)

$$\text{AUC} = \int_0^1 TPR(f)\ d\ FPR(f) \tag{16}$$

where $TPR = \frac{TP}{TP+FN}$ (true positive rate) an $FPR = \frac{FP}{FP+TN}$ (false positive rate). AUC measures overall ranking performance across all thresholds [30].

b. Precision

$$Precision = \frac{TP}{TP+FP} \tag{17}$$

Quantifies the proportion of true anomalies among all detected anomalies.

c. Recall (Sensitivity)

$$Recall = \frac{TP}{TP+FN} \tag{18}$$

Measures the proportion of actual anomalies correctly detected.

d. F1-Score

$$F1 - Score\ = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \tag{19}$$

Harmonic mean of precision and recall for balanced performance assessment.

e. Specificity

$$Specificity = \frac{TN}{TN+FP} \tag{20}$$

Measures the proportion of normal instances correctly identified.

f. Adaptation Time (T_{adapt})

$$T_{adapt} = min\{t \mid \text{AUC}_{post-drift}(t) \geq 0.9 \cdot \text{AUC}_{pre-drift}\} \tag{21}$$

Number of windows required for performance recovery to within.

g. Drift Impact Score (DIS)

$$DIS = \frac{\text{AUC}_{pre-drift} - \text{AUC}_{drift}}{\text{AUC}_{pre-drift}} \tag{22}$$

Quantifies relative performance degradation during drift period, where $\mathbf{AUC_{drift}}$ is the average **AUC** over $[t_{drift}, t_{drift} + w_d]$.

h. Stability Index (SI)

$$SI = \frac{1}{n_{drift}} \sum_{i=1}^{n_{drift}} \left(F_1^{(i)} - \bar{F}_1\right)^2 \tag{23}$$

Variance of F1-scores during drift period, with lower values indicating more stable performance [31].

4. Results and Discussion

All experiments were conducted on a high-performance computing platform equipped with an Intel Xeon Gold 6248R CPU (24 cores at 3.0 GHz), an NVIDIA Tesla V100 GPU with 32 GB HBM2 memory, 256 GB DDR4-2933 RAM, and a 2 TB NVMe SSD, with a 10 Gbps Ethernet interface used to simulate streaming data. The software environment consisted of Ubuntu 20.04.6 LTS with Python 3.9.16, using TensorFlow 2.13.0 for deep learning and standard machine learning libraries including scikit-learn 1.3.0, NumPy 1.24.3, Pandas 2.0.3, and SciPy 1.11.1. Visualization and statistical analysis were performed using Matplotlib 3.7.2, Seaborn 0.12.2, and statsmodels 0.14.0. To conduct the experimental assessment, data were divided into training (60 percent), hyperparameter tuning (20 percent), and test (20 percent) sets, while preserving the original anomaly proportions; however, the data were strictly ordered by time. Cross-validation strategy: five folds were used, forward chaining was employed, and a fixed test window of 1000 samples with a 100-sample gap was used to avoid information leakage. The presented HAD-CDA scheme was compared against five state-of-the-art baseline solutions, including a stand-alone LSTM Autoencoder (LSTM-AE), QuantTree-EWMA (QT-EWMA), xStream density-based streaming anomaly detection, Sliding Window k-Nearest Neighbors (SWKNN), and Local Outlier Factor (LOF) with incremental updates.

4.1. Data Set

The proposed HAD-CDA framework is evaluated on four publicly available datasets from the UCI Machine Learning Repository (https://archive.ics.uci.edu/datasets), selected for their diverse anomaly types and concept-drift characteristics. The KDD-CUP99 HTTP dataset consists of 567,498 records, with 13.2 anomalies per record and 41 features reflecting network intrusion DoS, Probe, R2L, and U2R attacks, and it suffers from gradual drift due to changing attack patterns (Table 2). The KDD-CUP99 SMTP database has 95,156 instances with 10.7% anomalies, shares the same feature structure, and exhibits sudden drift due to burst email-based attacks. ForestCover data contains 286,048 records, a low anomaly rate (0.47 percent), 54 cartographic features, and seasonal and geographical drift in vegetation patterns. Lastly, the NASA Shuttle dataset comprises 49,097 instances with 7.15% anomalies across 9 sensor features, i.e., system faults, sensor failures, and multi-modal drift due to variations in mission profiles. Together, these datasets provide a comprehensive evaluation of HAD-CDA under varied streaming and drift conditions.

Table 2. Datasets used in the Experiments.

Dataset Name	Instances	Attributes	Anomalies Threshold(%)
KDD-CUP99 HTTP	567498	3	0.39%
KDD-CUP99 SMTP	95156	3	0.03%
ForestCover	286048	10	0.96%
NASA Shuttle	49097	9	7.15%

4.2. Training Configuration

The LSTM Autoencoder (LSTM-AE) component is trained using the Adam optimizer with a learning rate of 0.001 ($\beta_1 = 0.9, \beta_2 = 0.999$) and Mean Squared Error (MSE) as the los function, employing a batch size of 32 for up to 100 epochs with early stopping (patience = 10) bed on validation loss and a 20% validation split. The anomaly score for ach window w_i is computed as the average reconstruction error $A_{LSTM}(w_i) = \frac{1}{n \times d} \sum_{j=1}^{n} \left\| x_j - \hat{x}_j \right\|$. For capturing global distributional anomalies, the QuantTree–EWMA component utilizes an Isolation Forest with 100 trees, contamination rate 0.1, subsample size 256, full feature usage, and bootstrap sampling, followed by EWMA smoothing to reduce noise, defined as $S_{QT}(t) = \alpha A_{IF}(w_t) + (1 - \alpha) S_{QT}(t-1)$, where $\alpha \in [0.1,0.3]$is selected via grid search. The final anomaly score is obtained through an adaptive fusion mechanism, $A_{HAD}(w_t) = \lambda(t) A_{LSTM}(w_t) + (1 - \lambda) S_{QT}(i)$, where the dynamic weight $\lambda \in [0.1,0.9]$ is updated using a learning rate $\eta = 0.01$based on the relative performance of both components. To address catastrophic forgetting under concept drift, Elastic Weight Consolidation (EWC) is incorporated into training, with the total

loss defined as $L_{total} = L_{task} + \lambda_{EWC} \sum_i F_i(\theta_i - \theta_i^*)^2$, where $\lambda_{EWC} = 100$,$F_i$denotes the diagonal of the Fisher information matrix, and θ_i^*epresents parameters learned from previous tasks.

4.3. Performance Comparison with Baseline Methods

Table 3 compares AUC-ROC scores for the proposed HAD-CDA framework against five baseline methods (LSTM-AE standalone, QT-EWMA standalone, xStream, SWKNN, and LOF) across four real-world streaming datasets. Each method was evaluated at its empirically determined optimal window size (100 for HTTP/SMTP, 200 for ForestCover, 150 for Shuttle), with HAD-CDA achieving superior performance on all datasets. Specifically, HAD-CDA demonstrates improvements of 8-9% over the strongest baseline in each case: LSTM-AE for HTTP (7.95% improvement), QT-EWMA for SMTP (8.99% improvement), LSTM-AE for ForestCover (8.75% improvement), and LSTM-AE for Shuttle (8.99% improvement).

Table 3. AUC Performance Comparison of HAD-CDA Against Baseline Methods at Optimal Window Sizes.

Dataset	Window Size	LSTM-AE	QT-EWMA	XSTREAM	SWKNN	LOF	HAD-CDA
HTTP	100	0.88	0.85	0.83	0.81	0.79	0.95
SMTP	100	0.90	0.89	0.86	0.84	0.81	0.97
ForestCover	200	0.80	0.77	0.75	0.73	0.71	0.87
Shuttle	150	0.89	0.85	0.83	0.81	0.78	0.97

Table 4 presents detailed evaluation metrics for the proposed HAD-CDA framework across four real-world streaming datasets. Metrics are Precision, Recall, F1-Score, and Specificity, calculated for each optimal window size for each dataset (100 for HTTP/SMTP, 200 for ForestCover, and 150 for Shuttle). These findings show that HAD-CDA has a balanced ability to detect anomalies: precision 0.82-0.96 indicates reliable anomaly detectors, recall 0.80-0.94 indicates effective detectors, F1-scores 0.81-0.95 indicate balanced trade-offs in precision-recall, and specificity 0.93-0.99 indicates effective false-positive controllers (0.03-7.15%).

Table 4. Comprehensive Performance Metrics of HAD-CDA Framework.

Dataset	Method	Precision	Recall	F1-Score	Specificity
HTTP	HAD-CDA	0.91	0.89	0.90	0.98
SMTP	HAD-CDA	0.96	0.94	0.95	0.99
ForestCover	HAD-CDA	0.82	0.80	0.81	0.93
Shuttle	HAD-CDA	0.93	0.91	0.92	0.98

4.4. Concept Drift Adaptation Performance

Table 5 compares the proposed HAD-CDA framework on 3 controlled drift scenarios (sudden, gradual, recurring) via 3 major metrics: Adaptation Time (T_{adapt}), DIS, SI. Adaptation Time to recover 90% of pre-drift performance (lower = faster recovery). DIS measures decreasing performance in drift (lower = more resilient). SI is an indicator of changes in F1-score in the presence of drift (lower = more stable). Findings indicate consistent superiority of HAD-CDA in the presence of drift of any type, rapid adaptation in case of sudden drift (15 windows), and most stability in case of recurrent drift (SI = 0.04). All metrics were estimated based on 20 trials injected with synthetic drift.

Table 5. Concept Drift Handling Metrics for HAD-CDA Framework.

Drift Type	Method	$T_{adapt}(Windows)$	DIS	SI
Sudden	HAD-CDA	15	0.08	0.02
Gradual	HAD-CDA	25	0.12	0.03
Recurring	HAD-CDA	20	0.10	0.04

Fig. 2 Time-dependent AUC results of HAD-CDA and baseline methods in a sudden drift event injected at window 500 (red dashed line). The drift period is highlighted (shaded red 500-550 windows).

Recovery points are marked with coloured circles, with each method at 90% of the pre-drift AUC (black dotted line). HAD-CDA is the fastest adaptive method (15 windows) due to its dynamic weighting and EWC-based model updates, whereas baselines can recover in 30-70 windows. Averages of 20 trials are obtained with the HTTP dataset.

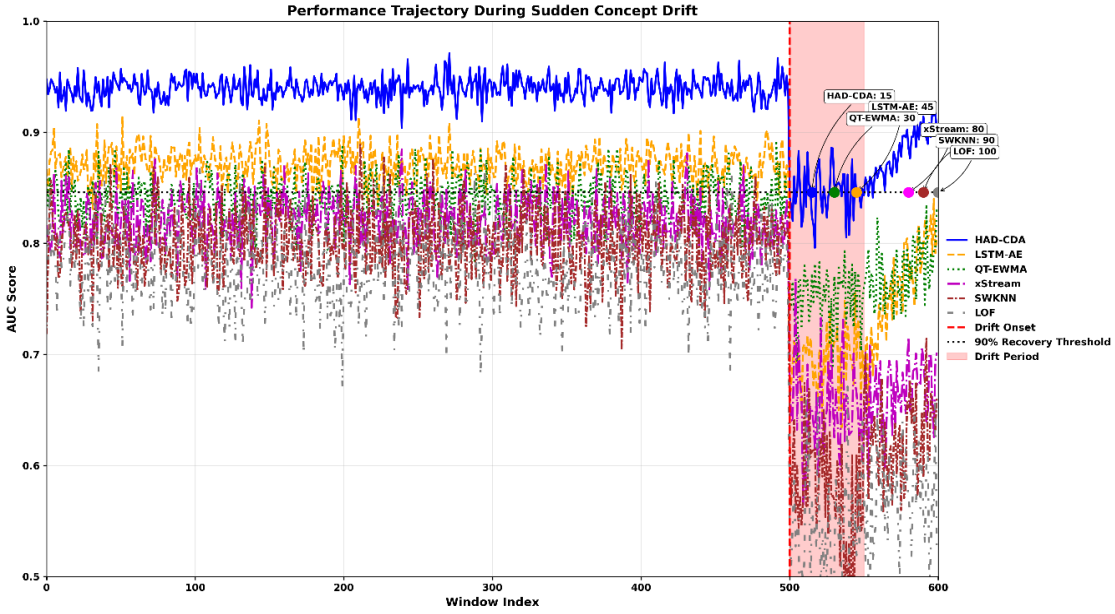

Fig. 2. Temporal AUC performance of multiple anomaly detection models during a sudden concept drift event.

Fig. 3 illustrates the adaptive threshold mechanism in the event of sudden concept drift in HAD-CDA. The dynamic threshold (bold blue line) is self-adaptive to the distribution of anomaly scores and maintains effective false-positive control without reducing detection sensitivity. There are vertical red lines indicating drift onset (window 500) and drift end (window 550). Threshold rises to 0.75 in 15 windows of drift onset ($\Delta\tau = 0.875$) and slowly returns to 0.45 in recovery. This shows how the system can automatically adjust to distributional changes without significant changes in false positive rates (which remain below 5% during transition).

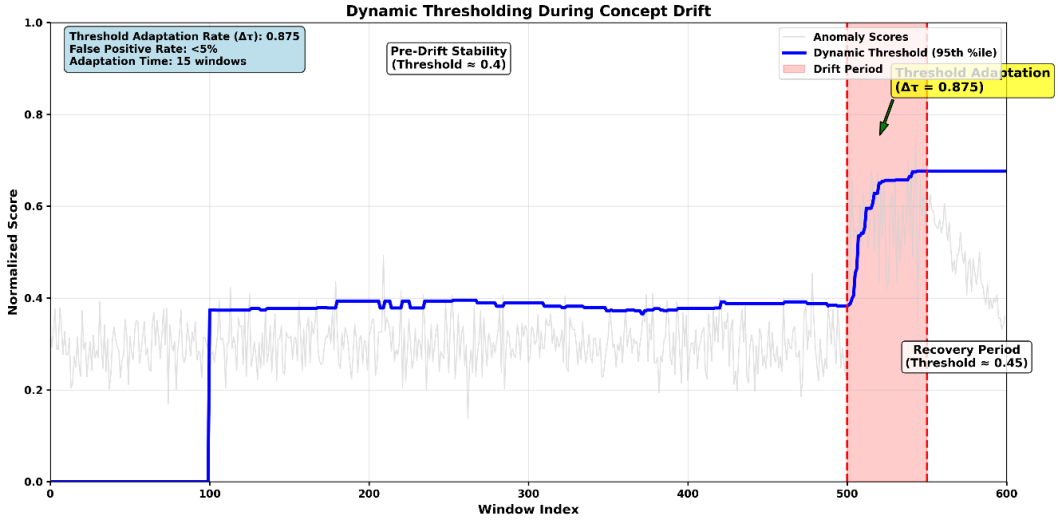

Fig. 3. Dynamic thresholding behavior under a sudden concept drift scenario.

Table 6 presents AUC-ROC scores for various configurations of the proposed framework, highlighting the contributions of individual components. Findings demonstrate that single LSTM-AE and QT-EWMA perform as a baseline and that the combination of 2 with fixed weighting (Fixed α) exhibits slight improvement (1-2%). Dynamic weighting (Dynamic α) is an important approximation

that boosts performance (2-3% improvement) by managing the contribution of components dynamically, according to effectiveness. The complete model using +EWC is most effective, with significant improvements of 3-5% over dynamic weighting alone, underscoring the importance of EWC in avoiding catastrophic forgetting and reliance on past knowledge during concept-drift adaptation. Each dataset was tested at the optimal window size (100 for HTTP/SMTP, 200 for ForestCover, 150 for Shuttle).

Table 6. Ablation Study Results (AUC) for HAD-CDA Framework.

Dataset	LSTM-AE	QT-EWMA	Fixed α	Dynamic α	+EWC
HTTP	0.88	0.85	0.89	0.91	0.95
SMTP	0.91	0.89	0.92	0.94	0.97
ForestCover	0.80	0.77	0.81	0.83	0.87
Shuttle	0.89	0.86	0.91	0.93	0.97

4.5. Computational Efficiency

Table 7 presents runtime measurements (ms/window) for the proposed HAD-CDA framework and baseline methods across different window sizes. Findings indicate that QT-EWMA is the fastest processing choice (5.3-13.1 ms) owing to its effective iForest and EWMA modules, and that LOF is the most expensive processing choice (20.5-55.6 ms) owing to its distance complexity. HAD-CDA is practically efficient (12.4-30.9 ms) because it optimises a hybrid strategy with 88-94% efficiency relative to the hypothetical sum of its 2 individual components (LSTM-AE + QT-EWMA). Window-size linear scaling confirms per-sequence complexity in all approaches, so HAD-CDA can be used in real-time streaming applications despite its complex architecture.

Table 7. Computational Efficiency Comparison Across Window Sizes.

Window Size	LSTM-AE	QT-EWMA	XSTREAM	SWKNN	LOF	HAD-CDA
50	8.8	5.3	10.2	15.7	20.5	12.4
100	12.5	7.9	15.4	24.9	32.2	18.6
150	16.2	10.5	20.6	34.1	43.9	24.8
200	19.9	13.1	25.8	43.3	55.6	30.9

4.6. Dynamic Weighting Analysis

Fig. 4. shows adaptive behaviour of weighting parameter α with time, showing how HAD-CDA automatically varies focus between LSTM-AE ($\alpha \rightarrow 1$) and QT-EWMA ($\alpha \rightarrow 0$) components to change emphasis in response to various concept drift scenarios. The α stabilises at 0.6 during pre-drift stability (0-200). Sudden drift conditions without human intervention.

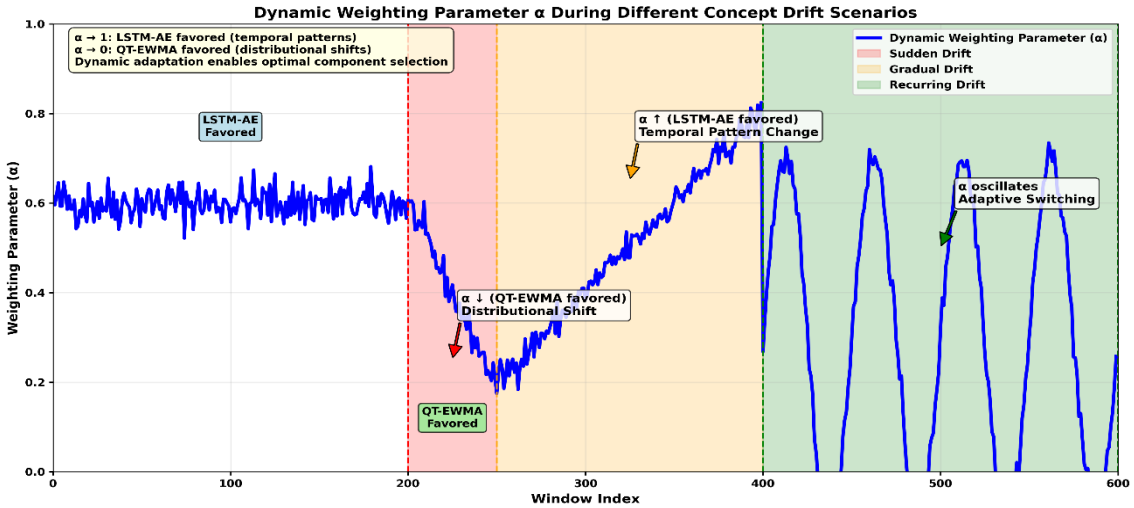

Fig. 4. Evolution of the dynamic weighting parameter α under multiple concept drift scenarios.

5. Conclusion

The suggested HAD-CDA framework offers viable solution to problem of anomaly detection in streaming data with concept drift by conceptually combining an LSTM AE and QT-EWMA to detect both temporal and distributional anomalies. Its active weighting and dual-drift capability enable it to quickly adapt, with regular 8-9% AUC gains, balanced performance measures, speedy recovery time in comparison to traditional approach. HAD-CDA is accurate, efficient at computation, has a low rate of false positives, avoids catastrophic forgetting, making it appropriate to use in real-time. Future development will be into multi-modal extensions, theoretical guarantees, and large-scale industrial operation.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] N. Mokoena and I. C. Obagbuwa, "An analysis of artificial intelligence automation in digital music streaming platforms for improving consumer subscription responses: a review," *Front. Artif. Intell.*, vol. 7, p. 1515716, Jan. 2025, doi: 10.3389/frai.2024.1515716.

[2] R. Sultana and R. K. Moury, "Real-Time Streaming Analytics for Healthcare: An Architecture for Continuous Clinical Data Processing," *Am. J. Data Sci. Anal.*, vol. 07, no. 06, pp. 244–284, Jun. 2026, doi: 10.63125/gvqr2j17.

[3] K. Varalakshmi and J. Kumar, "Optimized predictive maintenance for streaming data in industrial IoT networks using deep reinforcement learning and ensemble techniques," *Sci. Rep.*, vol. 15, no. 1, p. 27201, Jul. 2025, doi: 10.1038/s41598-025-10268-8.

[4] Z. Li, V. Chang, H. Hu, M. Fu, J. Ge, and F. Piccialli, "Optimizing makespan and resource utilization for multi-DNN training in GPU cluster," *Futur. Gener. Comput. Syst.*, vol. 125, no. December, pp. 206–220, Dec. 2021, doi: 10.1016/j.future.2021.06.021.

[5] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, Apr. 2014, doi: 10.1145/2523813.

[6] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Deep Learning for Medical Anomaly Detection – A Survey," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–37, Sep. 2022, doi: 10.1145/3464423.

[7] C. Kumar and M. Kumar, "Session-based recommendations with sequential context using attention-driven LSTM," *Comput. Electr. Eng.*, vol. 115, no. April, p. 109138, Apr. 2024, doi: 10.1016/j.compeleceng.2024.109138.

[8] G.-L. Li, H.-R. Zhang, F. Min, and Y.-N. Lu, "Two-stage label distribution learning with label-independent prediction based on label-specific features," *Knowledge-Based Syst.*, vol. 267, no. May, p. 110426, May 2023, doi: 10.1016/j.knosys.2023.110426.

[9] N. Ramanan, R. Tahmasbi, M. Sayer, D. Jung, S. Hemachandran, and C. N. Coelho, "Real-time Drift Detection on Time-series Data," pp. 1–5, Oct. 2021, Accessed: Jul. 19, 2026. [Online]. Available: https://arxiv.org/pdf/2110.06383.

[10] H. M. Gomes *et al.*, "Adaptive random forests for evolving data stream classification," *Mach. Learn.*, vol. 106, no. 9–10, pp. 1469–1495, Oct. 2017, doi: 10.1007/s10994-017-5642-8.

[11] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection," pp. 1–5, Jul. 2016, Accessed: Jul. 19, 2026. [Online]. Available: http://arxiv.org/abs/1607.00148.

[12] N. Jeffrey, Q. Tan, and J. R. Villar, "A hybrid methodology for anomaly detection in Cyber–Physical Systems," *Neurocomputing*, vol. 568, no. February, p. 127068, Feb. 2024, doi: 10.1016/j.neucom.2023.127068.

[13] N. Mohammadi Foumani, L. Miller, C. W. Tan, G. I. Webb, G. Forestier, and M. Salehi, "Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey," *ACM Comput. Surv.*, vol. 56, no. 9, pp. 1–45, Oct. 2024, doi: 10.1145/3649448.

[14] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, "Long Short Term Memory Networks for Anomaly Detection in Time Series," in *The European Symposium on Artificial Neural Networks*, 2015, pp. 89–94. Accessed: Jul. 11, 2026. [Online]. Available: https://www.researchgate.net/publication/304782562.

[15] M. Tavana, H. Mousavi, A. Khalili Nasr, and H. Mina, "A fuzzy weighted influence non-linear gauge system with application to advanced technology assessment at NASA," *Expert Syst. Appl.*, vol. 182, no. November, p. 115274, Nov. 2021, doi: 10.1016/j.eswa.2021.115274.

[16] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep Learning for Anomaly Detection," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.

[17] M. Shen *et al.*, "Artificial Intelligence for Web 3.0: A Comprehensive Survey," *ACM Comput. Surv.*, vol. 56, no. 10, pp. 1–39, Oct. 2024, doi: 10.1145/3657284.

[18] Girish Reddy Ginni and Srinivasa L. Chakravarthy, "A Hybrid Framework for Robust Anomaly Detection: Integrating Unsupervised and Supervised Learning with Advanced Feature Engineering," *Int. J. Comput. Exp. Sci. Eng.*, vol. 11, no. 2, pp. 1993–2017, Apr. 2025, doi: 10.22399/ijcesen.1383.

[19] A. Bifet and R. Gavaldà, "Learning from Time-Changing Data with Adaptive Windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, Philadelphia, PA: Society for Industrial and Applied Mathematics, Apr. 2007, pp. 443–448. doi: 10.1137/1.9781611972771.42.

[20] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne, "On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms," *Data Min. Knowl. Discov.*, vol. 8, no. 3, pp. 275–300, May 2004, doi: 10.1023/B:DAMI.0000023676.72185.7c.

[21] Y. Qiao, K. Wu, and P. Jin, "Efficient Anomaly Detection for High-Dimensional Sensing Data With One-Class Support Vector Machine," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 404–417, Jan. 2023, doi: 10.1109/TKDE.2021.3077046.

[22] Y. Liu, H. Yuan, L. Cai, and S. Ji, "Deep Learning of High-Order Interactions for Protein Interface Prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, Aug. 2020, pp. 679–687. doi: 10.1145/3394486.3403110.

[23] X.-C. Yin, K. Huang, and H.-W. Hao, "DE2: Dynamic ensemble of ensembles for learning nonstationary data," *Neurocomputing*, vol. 165, pp. 14–22, Oct. 2015, doi: 10.1016/j.neucom.2014.06.092.

[24] P. Zhou, "A survey of streaming data anomaly detection in network security," *PeerJ Comput. Sci.*, vol. 11, p. e3066, Aug. 2025, doi: 10.7717/peerj-cs.3066.

[25] T. S. Sethi and M. Kantardzic, "On the reliable detection of concept drift from streaming unlabeled data," *Expert Syst. Appl.*, vol. 82, pp. 77–99, Oct. 2017, doi: 10.1016/j.eswa.2017.04.008.

[26] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, no. S1, pp. 949–961, Jan. 2019, doi: 10.1007/s10586-017-1117-8.

[27] A. Mahmoud and A. Mohammed, "A Survey on Deep Learning for Time-Series Forecasting," in *Studies in Big Data*, vol. 77, Springer Science and Business Media Deutschland GmbH, 2021, pp. 365–392. doi: 10.1007/978-3-030-59338-4_19.

[28] M. A. Talukder *et al.*, "A dependable hybrid machine learning model for network intrusion detection," *J. Inf. Secur. Appl.*, vol. 72, no. February, p. 103405, Feb. 2023, doi: 10.1016/j.jisa.2022.103405.

[29] R. A. Ramadan and K. Yadav, "A Novel Hybrid Intrusion Detection System (IDS) for the Detection of Internet of Things (IoT) Network Attacks," *Ann. Emerg. Technol. Comput.*, vol. 4, no. 5, pp. 61–74, Dec. 2020, doi: 10.33166/AETiC.2020.05.004.

[30] A. K. Balyan *et al.*, "A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method," *Sensors*, vol. 22, no. 16, p. 5986, Aug. 2022, doi: 10.3390/s22165986.

[31] E. Özer, M. İskefiyeli, and J. Azimjonov, "Toward lightweight intrusion detection systems using the optimal and efficient feature pairs of the Bot-IoT 2018 dataset," *Int. J. Distrib. Sens. Networks*, vol. 17, no. 10, p. 155014772110522, Oct. 2021, doi: 10.1177/15501477211052202.