THE JOURNAL OF APPLIED ENGINEERING AND TECHNOLOGIES
2026 Vol. 1, No. 3, ISSN (O): 3154-7877

Rare-Event Memory Updating for Low-Frequency Threat Discovery in Streaming Security Logs

Ka Wai Chan[1], Mei Ling Ho[1], Jonathan Tsz Kit Leung[1*]

[1]Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong
Corresponding Author*: Jonathan Tsz Kit Leung E-mail: jtleung@cs.hku.hk

ARTICLE INFO

Keywords
Streaming security logs
Low-frequency anomaly detection
Online clustering
Cluster memory regulation
Log-template embedding
Concept drift
Security operations
Published:
02 September 2026

ABSTRACT

Security operation centers receive continuous log streams from firewalls, endpoint agents, authentication systems, cloud services, and intrusion prevention systems. Low-frequency anomalies are difficult to detect because they may appear as weak signals hidden among repeated benign events and gradually changing operational patterns. This study proposes a cluster memory regulation method for detecting low-frequency anomalies in streaming security logs. The method uses online clustering to summarize normal log-event behavior and introduces a memory-retention coefficient to preserve sensitivity to rare deviations. A log-template embedding module converts heterogeneous event messages into structured vectors, while a time-aware deviation score measures changes in event frequency, source identity, destination service, and action sequence. Experiments are conducted on an enterprise security-log stream containing 8.4 billion log entries collected from 18,200 endpoints, 1,140 servers, 76 firewall zones, and 320 cloud service accounts over 60 days. After template parsing, 4.9 million unique event sequences are mapped into streaming feature vectors. The dataset includes 3,760 annotated anomaly episodes, including abnormal login bursts, rare privilege-use sequences, suspicious outbound access, policy-denied activity surges, and endpoint process anomalies. The proposed method reduces median detection delay from 21.4 minutes to 6.8 minutes compared with a sliding-window isolation forest. False alarms are controlled at 5.2 investigation cases per 10 million log events. The system processes 156,000 parsed log events per second and maintains 12,600 active cluster states with 1.9 GB memory usage. Memory regulation prevents 1,430 rare anomaly patterns from being prematurely merged into normal clusters during long-running monitoring. The results demonstrate that cluster memory regulation can enhance streaming log anomaly detection under sparse and evolving threat conditions.

1. Introduction

Security operation centers (SOCs) continuously collect and analyze large volumes of logs generated by firewalls, endpoint devices, authentication systems, cloud platforms, intrusion prevention systems, network gateways, and business applications. These logs provide essential

Citation: Chan, K. W., Ho, M. L., & Leung, J. T. K. (2026). Rare-event memory updating for low-frequency threat discovery in streaming security logs. *The Journal of Applied Engineering and Technologies*, 1(3), 18-24.

3154-7877/© The Authors. Published by J&L Academic Group PLT. This is an open access article under the CC BY 4.0 license.
https://doi.org/10.64744/tjaet.2026.276.

evidence for identifying malicious activity, policy violations, compromised accounts, and abnormal communication behavior (Qi & Qiao, 2026; Jordan & Chen, 2025). However, enterprise security logs are highly heterogeneous and non-stationary. Their distributions may change because of software updates, user-role changes, network reconfiguration, cloud-resource migration, new access policies, and variations in daily workloads. As a result, a log pattern that is unusual in one operational context may be normal in another, making static detection rules difficult to maintain over long monitoring periods. Low-frequency security anomalies are particularly difficult to identify because they usually appear as weak and isolated deviations within a large number of repetitive normal events (Khotimah & Maharjan, 2025; Büyükbıçakcı, 2025). Rare privilege escalation, abnormal administrative access, unusual outbound connections, infrequent lateral movement, and atypical resource requests may occur only a few times before a serious incident develops (Su & Dong, 2026). Recent multivariate time-series research has shown that causal inference can model dependencies among variables, distinguish initiating abnormal factors from propagated effects, and improve fine-grained anomaly localization (Chen et al., 2025). This perspective is relevant to security-log monitoring because a single suspicious action may trigger multiple downstream events across authentication, endpoint, network, and cloud systems. Nevertheless, causal relationships estimated from historical data may become unreliable when system configurations and user behavior change. Rare threat signals may also be obscured by legitimate maintenance activity, automated scripts, service-account operations, and temporary workload fluctuations (Iyer, 2025; Yin et al., 2026).

Traditional security-monitoring methods commonly rely on sliding windows, fixed thresholds, rule matching, statistical isolation, or manually defined indicators. These methods are relatively easy to implement and interpret, but their effectiveness depends heavily on stable event distributions and carefully selected parameters (Zhang, 2026; Skliarov et al., 2025). A short sliding window may miss slowly developing attack behavior, whereas a long window may introduce excessive historical noise and delay detection (Zhang et al., 2026; Hong et al., 2026). Isolation-based methods can identify globally unusual observations, but they may perform poorly when malicious events are only slightly different from normal activity or when legitimate events are naturally sparse. Fixed thresholds may also generate large numbers of false alarms when normal access frequency, traffic volume, or user behavior changes. Recent studies have investigated online clustering, temporal embeddings, adaptive thresholding, sequence modeling, and autoencoder-based detection for streaming security logs (Li et al., 2026; Subhan et al., 2026). Micro-cluster and density-based approaches can summarize frequently occurring log patterns and update the monitoring model as new events arrive (Qi & Qiao, 2026; Su & Zhang, n.d.). Temporal representation methods can capture sequential relationships among authentication attempts, process activities, network connections, and access-control decisions. Autoencoders and other deep models can further learn nonlinear dependencies that are difficult to represent using manually designed rules (Rezaei et al., 2025; Du et al., 2026). However, many deep models require extensive offline training, substantial computing resources, and sufficiently representative historical data. Their performance may deteriorate when previously unseen applications, users, devices, or attack behaviors appear in the stream. Online clustering methods are more suitable for continuous processing because they incrementally update compact summaries without repeatedly retraining the entire model (Zhang et al., 2026; Bidaki et al., 2025). However, rapid updating can create a pattern-absorption problem. When a rare deviation occurs repeatedly over an extended period, the clustering model may gradually incorporate it into a normal micro-cluster. This process reduces the distance between abnormal and normal patterns and weakens sensitivity to persistent low-

frequency threats (Du et al., 2026; Xiong et al., 2026). Conversely, restricting model updates too strongly may prevent the detector from adapting to legitimate operational changes and increase false alarms. Effective long-term monitoring therefore requires a balance between adaptation to evolving normal behavior and retention of weak anomaly evidence (Slim et al., 2026; Veesam et al., 2025). Existing studies also frequently evaluate models using precision, recall, F1-score, or area-under-curve measures, while giving limited attention to detection delay, false alerts per monitored asset, event-processing throughput, cluster stability, and memory consumption. These operational indicators are critical for practical SOC deployment because even a statistically accurate model may be unsuitable if it generates excessive alerts or cannot process enterprise-scale streams in real time.

This study proposes a cluster memory regulation method for detecting low-frequency anomalies in continuous security-log streams. The proposed method introduces a memory-retention coefficient that controls the updating of cluster representations and prevents rare deviation patterns from being prematurely absorbed into normal clusters. Heterogeneous log messages are transformed into structured stream vectors through log-template embeddings, while event attributes related to frequency, source, destination, user identity, action type, and action sequence are incorporated into a time-aware deviation score. The method is evaluated on an enterprise-scale stream containing approximately 8.4 billion log entries, with experiments designed to examine detection delay, false-alarm frequency, cluster stability, processing throughput, and memory cost. The study aims to preserve low-frequency threat evidence while maintaining sufficient adaptation to legitimate changes in enterprise systems. By addressing the conflict between rapid cluster updating and long-term anomaly retention, the proposed method provides a scalable monitoring approach for identifying rare privilege misuse, abnormal outbound access, lateral movement, and other weak security signals. The findings are expected to improve early threat detection, reduce unnecessary analyst workload, support more reliable incident prioritization, and strengthen the long-term operational effectiveness of SOC monitoring systems.

2. Materials and Methods

2.1 Samples and Study Area

This study used streaming security logs from a large enterprise network. The dataset included 8.4 billion log entries from 18,200 endpoints, 1,140 servers, 76 firewall zones, and 320 cloud service accounts over 60 days. After log-template parsing, 4.9 million unique event sequences were converted into structured feature vectors. The dataset included normal operations, gradual changes in log patterns, repeated benign events, and 3,760 labeled low-frequency anomaly episodes, such as abnormal login bursts, rare privilege-use sequences, suspicious outbound access, policy-denied activity surges, and endpoint process anomalies. Data were collected continuously at sub-minute resolution and verified against system monitoring and security incident records.

2.2 Experimental Design and Control Experiments

The experimental group used the proposed cluster memory regulation method. This method keeps compact micro-clusters for normal log patterns and applies a memory-retention coefficient to prevent rare anomalies from being absorbed too quickly. Log-template embeddings convert heterogeneous log messages into structured vectors, and time-aware deviation scores detect unusual behavior. Control experiments included a sliding-window isolation forest, standard DenStream, and incremental k-means. All methods processed the same chronological log stream and were evaluated

using identical anomaly labels. This design tests whether memory regulation and temporal embeddings improve detection of low-frequency anomalies while maintaining stable behavior during normal operational drift.

2.3 Measurement Methods and Quality Control

Logs were collected automatically from endpoints, servers, firewall devices, and cloud services. Each entry included a timestamp, event type, source and destination identifiers, action type, and device status. Records with missing timestamps, duplicate entries, or invalid IDs were removed. Short gaps were filled using linear interpolation, and longer gaps were excluded. To ensure data quality, 5% of logs were randomly sampled and cross-checked with system monitoring data. The mismatch rate was below 1.5%. All feature vectors were normalized by log type to reduce scale differences and ensure consistent clustering across devices and events.

2.4 Data Processing and Model Formulation

Log streams were processed in time order using fixed-size online windows. Let x_t^i denote the feature vector of event sequence i at time t, and $C_j(t)$ the j-th micro-cluster. The distance between a new observation and its nearest cluster is:

$$d_t^i = \min_j \| x_t^i - \mu_j(t) \|_2,$$

where $\mu_j(t)$ is the center of cluster $C_j(t)$. To avoid absorbing repeated rare deviations too quickly, the cluster update weight is controlled by a memory-retention term:

$$\alpha_t = \alpha_0 \exp(-\lambda r_t),$$

where α_0 is the initial update rate, λ is the decay coefficient, and r_t is the recent repetition count of suspicious deviations. The cluster center is updated as $\mu_j(t+1) = (1-\alpha_t)\mu_j(t) + \alpha_t x_t^i$. Observations exceeding the temporal deviation threshold and not assignable to any normal micro-cluster are flagged as anomalies.

2.5 Statistical Analysis and Evaluation Metrics

Performance was evaluated using median detection delay, false alarms per 10 million log events, Matthews correlation coefficient, processing speed, memory use, and normal-cluster displacement. Detection delay is the time between the start of an annotated anomaly and the first alarm. False alarms were calculated during normal operations and normalized by total event count. The Matthews correlation coefficient measures performance under imbalanced anomaly rates. Cluster displacement quantifies the average shift of normal-cluster centers during gradual drift. Processing speed and memory use were recorded to assess feasibility for online deployment. Differences between the proposed method and baseline methods were tested using paired tests with .

3. Results and Discussion

3.1 Detection Performance for Low-Frequency Security Anomalies

The proposed cluster memory regulation method outperformed sliding-window isolation forest, standard DenStream, and incremental k-means models. Median detection delay decreased from 21.4 min to 6.8 min, indicating that rare abnormal events were detected earlier in streaming

security logs. The improvement comes from retaining memory for rare deviations rather than merging them into normal clusters too quickly (Zhang et al., 2026; Xiong & Zeng, 2026). In contrast, sliding-window methods rely on recent observations, which can cause rare anomalies to be treated as noise. As shown in Fig. 1, streaming log detection benefits from combining online clustering with memory regulation to maintain sensitivity to low-frequency events (Zhang et al., 2025; Kahoul et al., 2025).

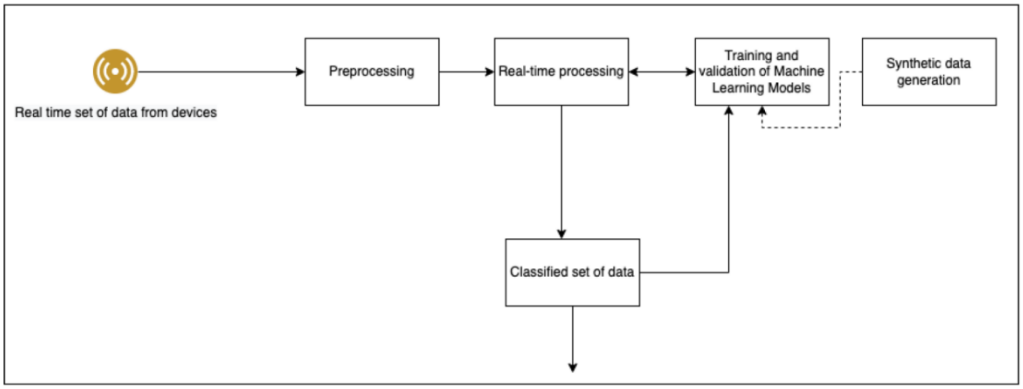

Fig. 1. Workflow of the proposed cluster memory regulation method for streaming security log detection.

3.2 False Alarm Control and Memory Retention

False alarms were reduced to 5.2 cases per 10 million log events, which is practical for security operation centers because high alert rates increase analyst workload. Memory retention prevented 1,430 rare anomaly patterns from being absorbed into normal clusters during long monitoring periods. Low-frequency events, such as rare privilege-use sequences or unusual outbound access, often reappear sparsely; standard clustering may gradually treat these as normal. By slowing cluster updates for repeated deviations, the proposed method preserves anomaly visibility while adapting to normal changes (Hong et al., 2026; Li et al., 2026).

3.3 Detection of Different Log Anomaly Types

The method effectively detected abnormal login bursts, rare privilege-use sequences, suspicious outbound access, policy-denied activity surges, and endpoint process anomalies. These events vary in duration and signal strength. Login bursts and denied-activity surges appear as short frequency changes, while privilege-use sequences and outbound anomalies involve source identity, destination, and action sequence. Log-template embeddings converted heterogeneous messages into structured vectors, enabling stable clustering (Alzahrani, 2026; Yin et al., 2026). Compared with methods relying on event counts or fixed thresholds, the time-aware deviation score captures changes in frequency, source, destination, and action order simultaneously. Fig. 2 shows that controlling false positives is critical in practical firewall-log anomaly detection.

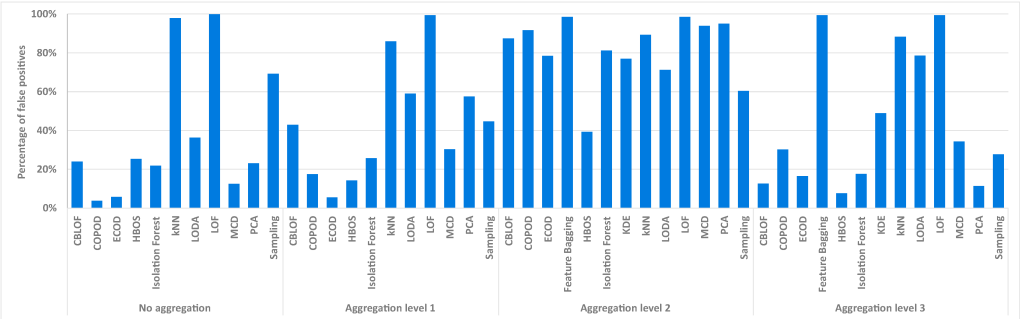

Fig. 2. False-positive comparison of different anomaly detection methods for security logs.

3.4 Comparison with Existing Studies and Practical Value

Compared with prior streaming log studies, this study used a large-scale, operational dataset including 8.4 billion log entries from 18,200 endpoints, 1,140 servers, 76 firewall zones, and 320 cloud accounts over 60 days. After parsing, 4.9 million unique event sequences were converted into streaming feature vectors. The online module processed 156,000 events per second and maintained 12,600 active cluster states using 1.9 GB of memory. These results show that the method can support large-scale security monitoring, reducing detection delay, limiting false alarms, and preserving sensitivity to rare anomalies. The study depends on labeled historical events, and future work should extend the method to longer periods, more attack types, cloud-native logs, and analyst feedback.

4. Conclusion

This study introduced a cluster memory regulation method for detecting low-frequency anomalies in streaming security logs. The method uses online clustering to summarize normal log patterns and applies a memory-retention coefficient to prevent rare deviations from being quickly merged into normal clusters. Log-template embeddings and a time-aware deviation score capture changes in event frequency, source, destination, and action sequence. Compared with sliding-window isolation forest and other baseline methods, the proposed method achieved shorter detection delay, fewer false alarms, stable memory usage, and improved detection of rare anomaly episodes. These results demonstrate that memory-controlled clustering can enhance long-term security monitoring, helping security operation centers detect threats early, prioritize investigations, and manage log data efficiently. Limitations include reliance on labeled anomaly episodes from historical records. Future work should extend the method to longer monitoring periods, additional attack types, cloud-native logs, and incorporate analyst feedback.

Acknowledgments

We are grateful to all respondents who participated in this study.

References

1. Alzahrani, M. (2026). Investigating the Impact of Log-Sequence Embeddings on Anomaly Detection: A Systematic Study. Information, 17(3), 228.
2. Bidaki, S. A., Mohammadkhah, A., Rezaee, K., Hassani, F., Eskandari, S., Salahi, M., & Ghassemi, M. M. (2025). Online continual learning: A systematic literature review of approaches, challenges, and benchmarks. arXiv preprint arXiv:2501.04897.
3. Büyükbıçakcı, E. (2025). Anomaly Detection in Salesforce's Transactional Data Using Machine Learning Techniques. Journal of Advanced Applied Sciences, 4(1-2), 1-15.
4. Chen, X., Xiao, H., Zeng, Z., Zhang, S., & Xiao, R. (2025). Fine-Grained Multivariate Time Series Anomaly Detection via Causal Inference. Knowledge-Based Systems, 114765.
5. Du, Y., Chen, Z., Liu, Q., & Dong, Y. (2026). Co-identification of Vibration, Structural Looseness, and Surface Damage in Industrial Equipment Operation Videos.
6. Du, Y., Dong, Y., Liu, Q., & Chen, Z. (2026). Modal Reliability Assessment and Drift Early Warning in Visible-Infrared Target Tracking. Available at SSRN 7132381.
7. Hong, Z., Chen, H., & Xu, T. (2026). Real-Time Capacity Risk Forecasting for Low-Latency Financial Trading Cloud Platforms. Available at SSRN 7118180.
8. Hong, Z., Xu, T., & Chen, H. (2026). A Study on Test Coverage Mapping and Release Risk

Prediction in Continuous Delivery of Cloud Services.
9. Iyer, K. I. (2025). From logs to intelligence: Leveraging data science for service account monitoring. Computer Fraud and Security, 1202-1209.
10. Jordan, A., & Chen, Y. (2025, October). User and Entity Behavior Analytics (UEBA) Enhanced Security Anomaly Detection in Enterprise DevSecOps Platforms. In 2025 IEEE Secure Development Conference (SecDev) (pp. 94-104). IEEE.
11. Kahoul, A. E. M., Bouraoui, Z., Zarour, K., & Boumezbeur, I. (2025). Hybrid deep learning ensemble with dynamic fusion for reliable anomaly detection in operational LTE RANs. Journal of Telecommunications and the Digital Economy, 13(4), 105-126.
12. Khotimah, K., & Maharjan, R. (2025). Improving Detection Accuracy of Brute-Force Attacks on MariaDB Using Standard Isolation Forest: A Comparative Analysis with RotatedVariant. MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, 25(1), 145-160.
13. Li, J., Ma, R., & Gu, Y. (2026). From Audit Requirements to Computable Software Architecture: A Rule-Engine and Evidence-Indexing Method for Trusted Digital Infrastructure.
14. Li, J., Zhang, Y., & Gu, Y. (2026). Chain-Hash Audit Framework with Trusted Anchoring for Tamper-Evident Evidence Lifecycle Management.
15. Qi, C., & Qiao, X. (2026). Building Reliable AI Systems: A Framework That Bridges Architecture and Operations. Available at SSRN 6795298.
16. Qi, C., & Qiao, X. (2026). Efficient Data Sampling and Feature Selection Algorithms for Scalable Machine Learning Pipelines.
17. Rezaei, Z., Sajedi, H., & Hashemi, M. (2025). Data stream clustering with concept drift using fractal dimension: Z. Rezaei et al. Knowledge and Information Systems, 67(10), 8715-8749.
18. Skliarov, M., Shawi, R. E., Dhaoui, C., & Ahmed, N. (2025). A comparative evaluation of explainability techniques for image data. Scientific Reports, 15(1), 41898.
19. Slim, A., Charef, M., & Maazouzi, F. (2026). IGF-Net: Adaptive Gated Fusion and Identity Mapping Framework with Local–Global Experts for Reconstruction Saturation Mitigation in Insider Threat Detection. Arabian Journal for Science and Engineering, 1-17.
20. Su, D., & Dong, Y. (2026). Classroom-Based Assessment with Bayesian Learning Analytics for Instructional Decision-Making in ASD Inclusive Education.
21. Su, D., & Zhang, H. Developing a Data-Driven Computational Framework for Scalable Special Education Practices for Autism and Intellectual Disabilities in the US Public School System.
22. Subhan, R. M., Lee, Y. D., & Koo, I. (2026). Autoencoder-based self-supervised anomaly detection in wireless sensor networks: a taxonomy-driven meta-synthesis. Applied Sciences, 16(3), 1448.
23. Veesam, S. B., Rao, B. T., Begum, Z., Patibandla, R. L., Dcosta, A. A., Bansal, S., ... & Al-Mugren, K. S. (2025). Multi-camera spatiotemporal deep learning framework for real-time abnormal behavior detection in dense urban environments. Scientific Reports, 15(1), 26813.
24. Xiong, W., & Zeng, Y. (2026). Multi-Layer Coupled Diagnosis of Energy Efficiency Degradation in Complex Building MEP Systems and Identification of Optimization Boundaries. Available at SSRN 7121782.
25. Xiong, W., Zeng, Y., & Guo, Y. (2026). A Study on the Characterization of Building MEP System States and the Learning of Behavioral Patterns Driven by Large-Scale Operational Data. Available at SSRN 7121883.
26. Yin, J., Huang, Y., & Rao, H. (2026). A Study on the Dynamic Evolution of Learning Behavior and Outcome Prediction in Digital Educational Environments. Available at SSRN 6607139.
27. Yin, J., Rao, H., & Huang, Y. (2026, March). Dynamic Modeling and Heterogeneity Analysis of Platform User Behavior Time Series. In 2026 International Conference on AI in Education Technology and Applications (AIETA) (pp. 30-33). IEEE.
28. Zhang, Y., Gu, W., & Wang, J. (2025, December). Research on First Article Inspection (FAI)-Driven Quality Assurance Methods for Wind Turbine Installation and Operation & Maintenance and Their Effect on Reliability Improvement. In Proceedings of the 2025 6th International Conference on Computer Science and Management Technology (pp. 1700-1705).

29. Zhang, Y., Wang, J., & Gu, W. (2026, February). Research on the Construction and Effectiveness of a Computer-Aided Systematic Training Framework for Wind Farm O&M Quality Amidst High Personnel Turnover. In Proceedings of the 2nd International Conference on Digital Management and Information Technology (pp. 647-651).
30. Zhang, Z. (2026). A Study on Multimodal Product Understanding and Assembly Guidance Optimization in e-commerce for Complex Consumer Goods. Available at SSRN 6734559.
31. Zhang, Z., Gao, Y., & Tong, Y. (2026). Semantic-Temporal Graph Learning for Demand Forecasting and Resource Allocation in Public Information Systems. Available at SSRN 6792279.
32. Zhang, Z., Tong, Y., & Gao, Y. (2026). Retrieval-Augmented Generation with Low-Latency Deployment for Vertical Domains Question Answering: A Case Study on Economic Resource Platforms.