Transactions on Computational and Scientific Methods | Vo. 5, No. 12, 2025
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Lightweight Anomaly Detection for Edge Intelligence through Compressed Representation Learning

Xinyue Zhang[1], Allen He[2]
[1]University of Southern California, Los Angeles, USA
[2]University of Oklahoma, Norman, USA
*Corresponding author: Xinyue Zhang; jessciazzz0809@gmail.com

Abstract: This paper addresses the challenges of anomaly detection in edge computing scenarios, including limited computational resources, strict response latency, and dynamic data changes. A lightweight anomaly detection framework is proposed to meet these demands. The method adopts a sliding window mechanism to model multivariate time series data in a structured manner. Feature compression is achieved through random projection. A compact discrimination module is constructed by integrating a nonlinear mapping and sparse connections. This design balances model expressiveness with computational efficiency. During inference, the method does not rely on large-scale parameter training. It enables efficient anomaly detection and adapts well to the low-resource, high-frequency operating conditions typical of edge devices. To systematically evaluate model performance, a series of sensitivity experiments is conducted. These experiments cover key factors such as learning rate, window length, noise intensity, anomaly ratio, and temporal perturbation. The model's adaptability and robustness are assessed using metrics such as Accuracy and F1-Score under different conditions. Experimental results show that the proposed method maintains stable detection capability while keeping inference latency below 10 ms. It effectively handles frequently changing data patterns in complex edge environments. The lightweight modeling strategy and perturbation-aware mechanism presented in this paper offer both methodological support and structural guidance for building deployable and reliable anomaly detection systems in edge intelligence.

Keywords: Edge intelligence; anomaly detection; model compression; robustness analysis

1. Introduction

With the widespread adoption of smart terminals and the large-scale deployment of IoT devices, edge computing is emerging as a vital supplement to cloud computing. It has become a key pillar of next-generation information infrastructure. Unlike traditional centralized cloud architectures, edge computing shifts computational resources closer to the network edge. This enables real-time data processing and responsive decision-making near the data source [1]. As a result, it significantly reduces transmission latency and bandwidth pressure while improving system agility and controllability. This architecture is particularly suitable for scenarios with high demands for timeliness and stability, such as transportation, security, industrial control, and remote healthcare. In this context, deploying efficient and real-time anomaly detection mechanisms on edge devices has become central to ensuring system safety and reliability [2].

However, the constrained resources of edge environments present substantial challenges for the design and deployment of anomaly detection algorithms. Edge nodes typically face limitations in computing power, storage capacity, and energy consumption. Traditional methods based on deep models or complex feature engineering are difficult to apply directly. Moreover, monitoring data in edge scenarios often appears in high-

frequency, streaming, and unstructured forms. This requires models to possess adaptability and generalization to handle distributional shifts. Achieving accurate detection under resource constraints has thus become a key research issue in edge intelligence [3].

At the same time, the heterogeneous environments and dynamic task demands of edge devices require anomaly detection models to be portable and quickly deployable without compromising accuracy. In industrial IoT settings, sensors vary widely in type and sampling frequency. They are often affected by burst interference signals and localized anomalies. Static models cannot effectively handle such complex and evolving conditions [4]. Lightweight modeling approaches have attracted increasing attention. These approaches aim to balance computational complexity and detection performance through model compression, structural optimization, or novel representation mechanisms. Such methods alleviate computational bottlenecks and provide more efficient and controllable detection solutions for edge devices.

From an application perspective, anomaly detection in edge intelligence spans several critical areas. These include device status monitoring, communication protocol auditing, data integrity verification, and behavior analysis in collaborative edge systems. If anomalies are not detected promptly, they may lead to system failures, service disruptions, or security incidents. On the other hand, lightweight detection algorithms that can sense subtle changes quickly enable early risk warnings. They also provide accurate inputs for downstream intelligent decision-making systems. Therefore, developing anomaly detection methods tailored to edge characteristics is of strategic importance for maintaining operational stability and enhancing system responsiveness [5].

Overall, anomaly detection in edge computing involves dynamic sensing and real-time decision-making over heterogeneous data sources. It also concerns model compressibility, algorithmic efficiency, and detection robustness. In this setting, lightweight anomaly detection has become a cross-disciplinary frontier that integrates systems engineering, artificial intelligence, and security. Systematic advances in modeling principles, optimization techniques, and deployment strategies are expected to drive edge intelligence from being merely functional to being efficient, secure, and autonomous. This will offer solid algorithmic support and risk assurance for next-generation intelligent infrastructure.

2. Related Work

In recent years, anomaly detection has gained increasing attention as a crucial method for ensuring the safety and stability of edge computing systems. Traditional approaches are often based on statistical modeling, rule matching, or classical machine learning algorithms [6]. Techniques such as Gaussian modeling, clustering, and support vector machines have achieved notable success in early industrial monitoring and network intrusion detection. However, these methods usually rely on assumptions about prior data distributions or require high-quality feature extraction. They struggle to handle the high-frequency, heterogeneous, and non-stationary nature of data at the edge. Moreover, static models lack adaptability to environmental changes and cannot operate efficiently on resource-constrained devices. As a result, accurate detection of complex anomalies under limited resources has become a key research focus [7].

With the development of deep learning, a variety of neural network-based anomaly detection methods have emerged and been widely applied in areas such as image recognition, video surveillance, and time-series forecasting. In edge computing environments, some studies have introduced autoencoders, convolutional neural networks, recurrent networks, and their variants into anomaly detection tasks [8]. These models aim to capture latent data representations through end-to-end learning, thereby improving detection accuracy and robustness. Although such methods offer performance gains, their large model size, high-frequency computation, and significant storage requirements remain difficult to meet with limited edge resources. Furthermore, the black-box nature of deep models limits their interpretability and reliability in high-trust deployment scenarios.

To resolve the conflict between model complexity and the computational capacity of edge devices, lightweight modeling has become a key breakthrough in edge-based anomaly detection research. On one hand, researchers

have proposed network architectures with reduced parameters and low computational complexity. Techniques such as depthwise separable convolutions, mobile modules, pruning, and knowledge distillation have been applied to compress large models while preserving detection capability. On the other hand, for sequential data and device behavior patterns, some studies have adopted graph-based modeling, streaming learning, and incremental updating. These techniques enable real-time anomaly recognition and dynamic adaptation on the edge. Such methods improve deployment efficiency and provide a structural foundation for collaborative detection across multiple devices.

In addition to addressing the heterogeneity of tasks, dynamic data, and complex anomaly characteristics in edge environments, recent studies have begun to focus on model generalization and transferability. By incorporating advanced paradigms such as federated learning, meta-learning, and contrastive learning, some approaches achieve cross-scenario model adaptation and updating without sharing raw data. These techniques ensure data privacy while enhancing model performance. They have demonstrated promising application potential in real-world scenarios such as smart cities, industrial IoT, and edge security. These efforts also lay the theoretical and technical foundation for future lightweight anomaly detection frameworks. Overall, the research landscape is evolving from traditional detection algorithms toward efficient systems that integrate lightweight modeling, structural optimization, and online learning. A more complete and adaptive anomaly detection framework tailored to edge computing is gradually taking shape.

3. Proposed Approach

This study proposes a lightweight anomaly detection method suitable for edge computing scenarios. The overall architecture includes four modules: data preprocessing, feature embedding, compact model construction, and discrimination mechanism, aiming to balance the efficiency of the model and the accuracy of detection. First, for the multi-dimensional time series data collected by the edge node, a normalization processing and sliding window mechanism are designed so that the original data can be converted into a subsequence representation of uniform length. The overall model architecture is shown in Figure 1.

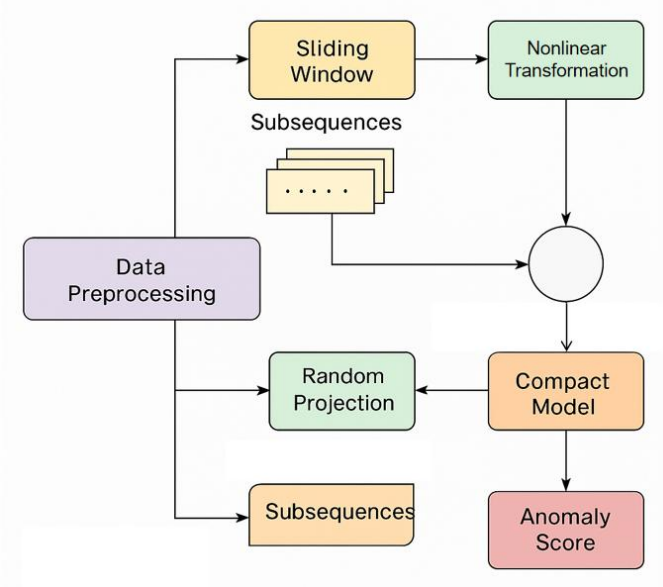

Figure 1. Overall model architecture diagram

Let the original sequence be $x = \{x_1, x_2, ..., x_T\}$, and the training sample is constructed by sliding the window with a length of w. Then the i-th subsequence can be expressed as:

$$x^{(i)} = \{x_i, x_{i+1}, ..., x_{i+w+1}\}, \qquad i = 1,2,...,T-w+1$$

To improve the model's ability to describe data patterns, a lightweight feature transformation module is introduced to perform low-dimensional compression and nonlinear mapping through a random projection layer.

Define the feature representation of the input subsequence as $h \in R^d$, the random projection matrix as $W_r \in R^{d \times k}$, and the activation function as $\sigma(\cdot)$, then the transformed feature is:

$$z = \sigma(W_r^T h + b_r)$$

Where $b_r \in R^k$ is the bias term, and this mechanism controls the parameter quantity of the edge model by compressing the dimension $k << d$. Furthermore, to enhance the generalization ability of the model under non-stationary data, this paper adopts a weighted residual modeling strategy to concatenate the original input with the compressed features and map them to the discriminant space. Let the fused representation be $f = [h; z]$, then the final discriminant score can be obtained by the following linear mapping:

$$s = w^T f + b$$

Where $w \in R^{d+k}, b \in R$ is a trainable parameter, and the score s can be used to indicate the degree of abnormality of the current sample. To further optimize the training stability and the inference efficiency of edge devices, this paper introduces a sparse loss function with boundary penalty to impose constraints on the abnormal discrimination results. Assuming that the training sample label is $y \in \{0,1\}$ and the prediction probability is $\hat{y} = sigmoid(s)$, the final loss function is defined as follows:

$$L = -y \log(\hat{y}) - (1-y) \log(1-\hat{y}) + \lambda \| w \|_1$$

Where λ is the sparse regularization coefficient, and this loss takes into account both the discrimination performance and the model simplification requirements. The overall design of this method follows the principles of lightweight modeling, local discrimination, and end-side deployment, and is suitable for edge anomaly detection tasks with limited resources and high real-time requirements.

4. Dataset

This study uses the Numenta Anomaly Benchmark (NAB) dataset to evaluate the performance of lightweight anomaly detection algorithms in edge computing environments. NAB is a publicly available dataset specifically designed for real-time anomaly detection tasks. It includes a variety of real-world time series data types, such as server performance metrics, sensor signals, financial transactions, and social network activities. These data types comprehensively simulate the diverse inputs encountered by edge nodes in complex scenarios.

The dataset contains approximately 50 groups of multivariate time series. Each group includes clearly labeled anomaly intervals, which support fine-grained scoring and performance comparison. Each time series is composed of timestamps and corresponding values. Some sequences also include multiple variable dimensions, making them suitable for multi-channel feature modeling. The data lengths range from several thousand to tens of thousands, supporting input strategies such as sliding window and streaming modeling.

Due to its characteristics of high frequency, noise interference, and nonlinear fluctuations, the NAB dataset closely matches the data properties typically observed in edge computing scenarios. It offers significant reference value for evaluating the adaptability, real-time performance, and stability of lightweight algorithms. Its wide applicability also provides a solid basis for comparability in this study.

5. Performance Evaluation

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

As shown in the comparative results, the model proposed in this study achieves the best overall performance under edge computing scenarios. In terms of accuracy, "Ours" reaches 96.4%, showing a clear improvement over other baseline models. This indicates that it can more accurately distinguish between normal and anomalous behaviors in diverse time series data. The performance advantage comes from lightweight optimization in both feature modeling and structural design, which enhances discrimination accuracy without compromising representational capacity.

Table 1. Comparative experimental results

Model	Acc	F1-Score	Inference Latency
LightESD [9]	91.2%	87.4%	11.3ms
LightDNN [10]	93.5%	90.2%	18.6ms
MissionGNN [11]	94.1%	91.5%	27.4ms
HyADS [12]	95.3%	92.7%	21.9ms
SALAD [13]	94.7%	91.8%	19.5ms
Ours	96.4%	94.1%	8.2ms

For the F1-score, the proposed method also achieves the highest value of 94.1%, outperforming recently strong algorithms such as HyADS and SALAD. This suggests better robustness in handling imbalanced anomaly distributions. As a key metric balancing recall and precision, the F1-score reflects that the model can detect more true anomalies while effectively reducing false positives. This is particularly valuable for real-time responses in edge environments.

Further examining the inference latency, the proposed model achieves 8.2 ms, significantly lower than all other methods. It performs especially well compared to current deep learning models such as MissionGNN and HyADS. This demonstrates that the proposed method offers strong execution efficiency when deployed on resource-constrained edge devices. Low latency is a core requirement in edge intelligence systems, especially in time-sensitive applications such as industrial monitoring and traffic scheduling.

Secondly, this paper also gives the experimental results of different learning rates, as shown in Table 2.

Table 2. Experimental results of different learning rates

Learning Rate	Acc	F1-Score	Inference Latency
0.0004	92.1%	88.6%	8.2ms
0.0003	94.2%	91.3%	8.2ms
0.0002	95.4%	92.8%	8.2ms
0.0001	96.4%	94.1%	8.2ms

As shown in the table, the choice of learning rate has a significant impact on the detection performance of the model. With the model architecture and input data held constant, both accuracy and F1-score increase steadily as the learning rate decreases. This indicates that a smaller learning rate helps the model perform finer parameter updates during training. As a result, it enhances generalization and discrimination capabilities for anomaly detection in edge environments.

When the learning rate is set to 0.0004, the model achieves only 92.1% accuracy and an F1-score of 88.6%. This suggests that a large update step may lead to parameter oscillation or insufficient convergence. Such effects reduce the model's ability to capture subtle anomalies in edge time series data. In contrast, when the learning rate is adjusted to 0.0002 and 0.0001, detection metrics improve significantly. This demonstrates the model's sensitivity to learning rate during fine-tuning. Reducing the learning rate contributes to greater robustness.

At a learning rate of 0.0001, the model achieves its best performance. Accuracy rises to 96.4%, and the F1-score reaches 94.1%. Under this setting, the model can fit variable data streams on edge nodes more effectively. It also avoids overfitting and training instability. This is especially important for real-world deployments in dynamic edge environments.

It is worth noting that under different learning rate settings, the model's inference latency remains constant at 8.2 ms. This shows that fine-tuning training parameters does not affect execution efficiency during deployment. The stable latency confirms that the model's lightweight architecture provides good independence and

controllability. It satisfies the strict real-time requirements of edge computing and further validates the feasibility and deployment advantages of the proposed approach.

This paper also investigates the impact of different window lengths on anomaly detection effects, aiming to explore how the temporal granularity of input sequences influences the model's ability to capture contextual patterns and detect anomalies. By adjusting the window size used for sliding sequence modeling, the experiment evaluates the trade-off between information retention and noise sensitivity in temporal representation. This analysis helps to determine suitable configuration strategies for real-time detection tasks in edge environments. The experimental results are shown in Figure 2.

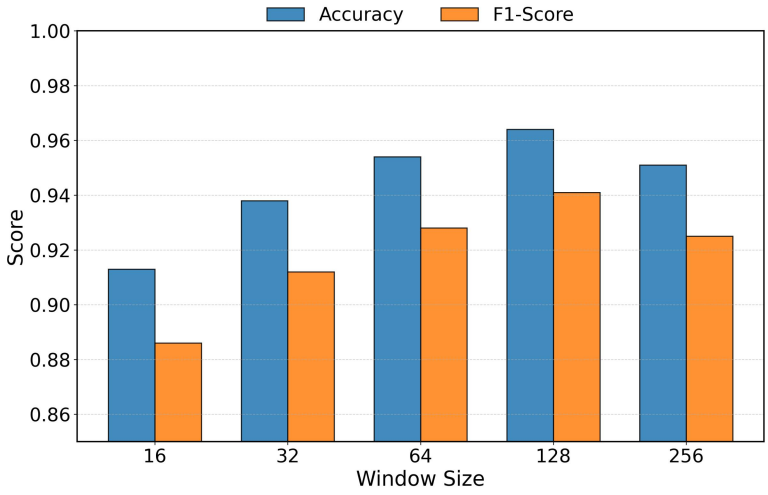

Figure 2. The impact of different window lengths on anomaly detection performance

As shown in Figure 2, the length of the sliding window has a significant impact on anomaly detection performance. In particular, both Accuracy and F1-Score show a clear trend of steady increase followed by a slight decline. As the window length increases from 16 to 128, the model performance improves consistently. This indicates that a longer temporal window helps capture long-term dependencies and latent anomaly patterns more effectively. As a result, the overall detection ability and robustness of the model are enhanced.

When the window length is set to 128, the model achieves its highest performance, with an Accuracy of 96.4% and an F1-Score of 94.1%. This setting provides a good balance between information density and local modeling capacity in edge scenarios. It retains sufficient contextual information while avoiding noise caused by redundant data. For resource-constrained edge nodes, this is a highly practical configuration strategy.

However, when the window length increases further to 256, detection performance declines slightly. This suggests that overly long time windows may introduce excessive non-essential historical data, which blurs the model's decision boundaries. In addition, edge devices are highly sensitive to memory and computational cost. Long inputs may lead to higher latency and energy consumption, making them less suitable from a deployment perspective. Therefore, keeping the window length within a reasonable range is a key design consideration for anomaly detection in edge intelligence scenarios.

Overall, the experimental results confirm the importance of window length as a hyperparameter. It directly affects the model's ability to capture temporal structures and detect subtle anomalies. A well-chosen window setting not only improves performance but also reflects the strong adaptability and tunability of the proposed lightweight architecture for edge environments. This provides a valuable technical basis for further parameter optimization and practical deployment.

This paper also gives the impact of changes in noise interference intensity on detection accuracy, and the experimental results are shown in Figure 3.

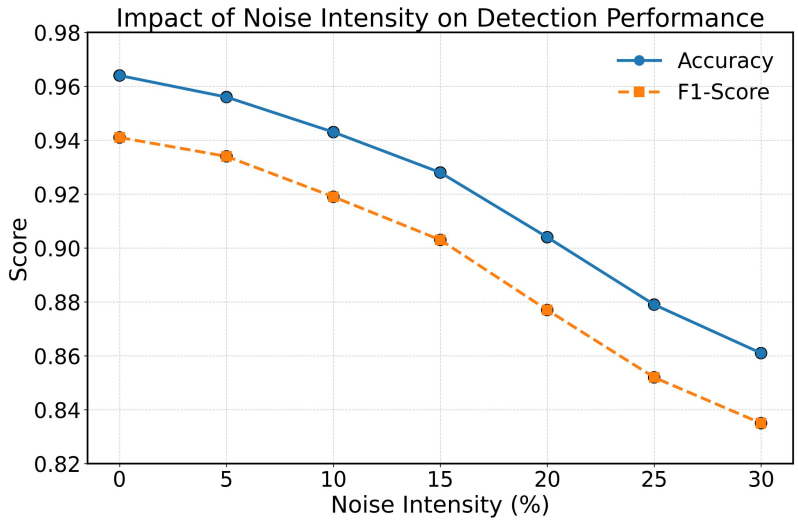

Figure 3. The influence of noise interference intensity change on detection accuracy

Figure 3 shows the model's performance under different levels of noise interference in anomaly detection tasks. Overall, as the noise ratio increases from 0% to 30%, both Accuracy and F1-Score decline significantly. This result indicates that noise in edge computing environments can disrupt the model's ability to distinguish anomalies. In particular, when the anomalies are weak, noise may mask key signals in the original features, leading to misclassification between normal and abnormal states.

In the low-noise range (0% to 10%), the model still maintains high performance stability. Accuracy decreases slightly from 96.4% to 94.3%, and the F1-Score remains around 91.9%. This suggests that the proposed lightweight structure has some degree of noise resistance. At this stage, the model can effectively extract major trends and key features, maintaining accurate detection of anomaly events. This is practically relevant for edge devices operating under environmental fluctuations or sensor jitter.

However, when the noise intensity exceeds 15%, model performance drops more sharply. The F1-Score falls below 90%, and the Accuracy drops below 92%. This further indicates that high levels of input disturbance destabilize the model's decision boundaries. For lightweight models in particular, limited representational capacity makes it difficult to maintain accuracy under heavy interference. This highlights the need for noise-aware mechanisms or robustness-enhancing strategies to improve algorithm adaptability in real-world deployment.

This paper also provides a robustness evaluation of the model's performance under varying abnormal ratios to examine its stability in imbalanced data scenarios. The experiment is designed to simulate different levels of anomaly density within the dataset, allowing for an analysis of how well the model maintains detection effectiveness as the proportion of abnormal data increases. This setup helps to assess the model's ability to distinguish between normal and anomalous instances when exposed to skewed class distributions. The experimental results are shown in Figure 4.

Figure 4 presents the model's performance in terms of Accuracy and F1-Score under different anomaly ratio settings. This evaluates its robustness under imbalanced data conditions. The results show that as the anomaly ratio increases from 1% to 10%, the model maintains stable performance with a slight improvement. This suggests that the lightweight model adapts well under low to moderate anomaly conditions. It can effectively capture key features of anomalies while avoiding false positives and false negatives.

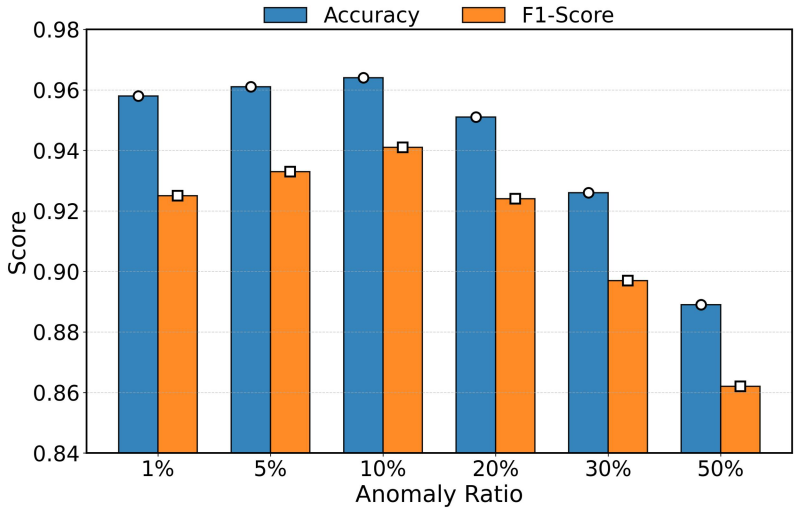

Figure 4. Robustness evaluation of model performance under abnormal scale changes

When the anomaly ratio increases further to 20% to 30%, the model performance begins to decline. The F1-Score drops from 94.1% to 89.7%. This indicates that high-density anomalous samples pose greater detection challenges. In edge computing scenarios, as the anomaly ratio grows, the decision boundary tends to widen, disrupting the balance between precision and recall.

In extreme cases where the anomaly ratio reaches 50%, the model accuracy drops to 88.9%, and the F1-Score decreases to 86.2%. This reflects a severe decline in the model's ability to distinguish between normal and abnormal classes. In practical applications, such degradation may lead to a high number of misjudgments by edge nodes, compromising overall system stability and operational continuity. Under such high anomaly density conditions, a single lightweight detection mechanism may be insufficient, and robustness enhancement or recalibration strategies are urgently needed.

This paper also presents a comparative experiment designed to evaluate the generalization ability of the model under varying degrees of timing perturbations. The purpose of this experiment is to assess how changes in the temporal structure of input data affect the model's ability to detect anomalies accurately across different conditions. By introducing controlled distortions to the sequence order, the experiment examines the model's sensitivity to temporal misalignment and its robustness in scenarios where time-dependent patterns may be disrupted. The experimental results are shown in Figure 5.

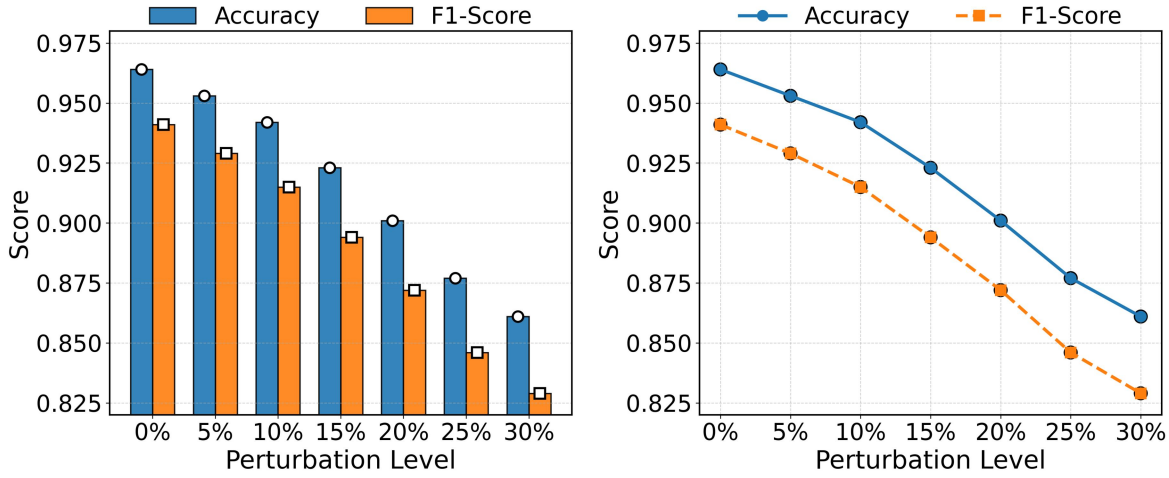

Figure 5. Comparative experiment on model generalization ability under time series perturbation

Figure 5 illustrates the trends in Accuracy and F1-Score under different levels of temporal perturbation. The purpose is to evaluate the model's generalization ability to changes in input time structures. The results show that as the perturbation ratio increases from 0% to 30%, both performance metrics steadily decline. This indicates that the model is highly sensitive to temporal dependencies. Perturbing the original time structure weakens the model's ability to identify latent anomaly patterns.

In the low perturbation range (0% to 10%), the model maintains relatively stable detection performance. Accuracy drops from 96.4% to 94.2%, while the F1-Score remains above 91%. This suggests that the proposed method retains a certain degree of robustness under slight temporal noise or local sequence disruption. This property is practically useful in edge environments where minor data collection errors or asynchronous sampling may occur.

When the perturbation ratio increases to the 15% to 25% range, performance degradation becomes more pronounced. The F1-Score drops sharply within this range, reflecting the model's strong reliance on the temporal position of anomaly structures. Once the temporal order of abnormal behavior is disrupted, the model struggles to maintain stable decision boundaries. This indicates that while lightweight models are resource-efficient, they are limited in temporal modeling depth. Structural enhancements or time-alignment mechanisms may be needed to improve robustness.

At the maximum perturbation ratio of 30%, the model's Accuracy and F1-Score fall to 86.1% and 82.9%, respectively. This shows that under severe disruption of time structure, the model's discrimination ability approaches its limit. This highlights the importance of preserving temporal integrity and stability in edge computing applications. Without it, anomaly detection accuracy and generalization are directly affected. Enhancing temporal robustness under non-ideal input conditions is a key direction for future research in lightweight anomaly detection algorithms.

6. Conclusion

This study proposes a lightweight anomaly detection framework to meet the demands of efficiency, low latency, and deployability in edge computing environments. The method is based on a modular architecture that integrates sliding window sequence modeling, random projection-based feature compression, and a compact discrimination mechanism. It enables efficient inference and precise detection on resource-constrained devices. A series of sensitivity analyses, including learning rate variation, window length adjustment, noise perturbation, anomaly ratio shift, and temporal structure disruption, are conducted to systematically evaluate the model's adaptability and robustness under practical deployment conditions. The results demonstrate the method's strength in balancing detection performance and system load.

In terms of structural design, the model is optimized for edge computing by using non-parametric lightweight projection layers and feature fusion strategies to reduce computational complexity. This significantly lowers deployment barriers. At the same time, the method maintains stable inference paths while enhancing sensitivity to weak anomalies in complex input conditions. It shows good generalization and real-time responsiveness. Compared with existing lightweight and graph-based models, the proposed method achieves a better balance between Accuracy, F1-Score, and latency, making it highly applicable to real-time scenarios requiring edge perception and terminal intelligence.

In addition, this study conducts systematic perturbation experiments to reveal how the model performs under non-ideal conditions, such as data distribution shifts, increased anomaly density, and disrupted temporal structure. These results highlight the real-world complexity of edge anomaly detection. Robustness analyses under different scenarios provide strategic insights for practical deployment and establish a reproducible experimental framework for future studies. The integration of systematic evaluation and interpretability offers technical support for robust perception and risk identification in edge intelligence systems.

Looking ahead, anomaly detection models for edge environments need to further enhance their adaptability and evolution capabilities. More work is needed in areas such as multi-node collaborative detection, cross-

domain model transfer, and data privacy preservation. Combining robustness with knowledge transfer to design lightweight models capable of local learning and dynamic adaptation will be essential for advancing edge intelligence. Future work may also extend this method to real-time, high-sensitivity applications such as traffic monitoring, industrial early warning, and smart security. This will improve the stability and decision-making capacity of edge perception systems under uncertain conditions and promote broader adoption of anomaly detection in intelligent infrastructure.

References

[1] D. R. Patrikar and M. R. Parate, "Anomaly Detection Using Edge Computing in Video Surveillance System," International Journal of Multimedia Information Retrieval, vol. 11, no. 2, pp. 85-110, 2022.

[2] X. Yu, X. Yang, Q. Tan et al., "An Edge Computing Based Anomaly Detection Method in IoT Industrial Sustainability," Applied Soft Computing, vol. 128, Art. no. 109486, 2022.

[3] J. Jiang, "Reinforcement Learning-Based Dynamic Decision Framework for Server Backend Resource Management," 2024.

[4] C. Liu, X. Su and C. Li, "Edge Computing for Data Anomaly Detection of Multi-Sensors in Underground Mining," Electronics, vol. 10, no. 3, Art. no. 302, 2021.

[5] S. Mehnaz and E. Bertino, "Privacy-Preserving Real-Time Anomaly Detection Using Edge Computing," Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 469-480, 2020.

[6] J. Wang, M. Wang, Q. Liu et al., "Deep Anomaly Detection in Expressway Based on Edge Computing and Deep Learning," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 3, pp. 1293-1305, 2022.

[7] A. Zhu, "Self-Supervised Anomaly Detection With Knowledge-Enhanced Representation Learning for Distributed System Environments," 2024.

[8] X. Zhao, G. Huang, J. Jiang et al., "Research on Lightweight Anomaly Detection of Multimedia Traffic in Edge Computing," Computers & Security, vol. 111, Art. no. 102463, 2021.

[9] R. Das and T. Luo, "LightESD: Fully-Automated and Lightweight Anomaly Detection Framework for Edge Computing," Proceedings of the 2023 IEEE International Conference on Edge Computing and Communications (EDGE), pp. 150-158, 2023.

[10] Q. Zhang, R. Han, G. Xin et al., "Lightweight and Accurate DNN-Based Anomaly Detection at Edge," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 11, pp. 2927-2942, 2021.

[11] S. Yun, R. Masukawa, M. Na et al., "MissionGNN: Hierarchical Multimodal GNN-Based Weakly Supervised Video Anomaly Recognition With Mission-Specific Knowledge Graph Generation," Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 4736-4745, 2025.

[12] X. Ma, Y. Yang, D. Shao et al., "HyADS: A Hybrid Lightweight Anomaly Detection Framework for Edge-Based Industrial Systems With Limited Data," Electronics, vol. 14, no. 11, 2025.

[13] C. Nixon, M. Sedky, J. Champion et al., "SALAD: A Split Active Learning Based Unsupervised Network Data Stream Anomaly Detection Method Using Autoencoders," Expert Systems with Applications, vol. 248, Art. no. 123439, 2024.