*Pinnacle Academic Press*
*Proceedings Series*

*Vol. 2 2025*

*Article* **Open Access**

# Improving Database Anomaly Detection Efficiency through Sample Difficulty Estimation

**Maoxi Li [1,*], Daobo Ma [2] and Yingqi Zhang [3]**

[1] Business Analytics, Fordham University, NY, USA
[2] Business Administration, California Institute of Advanced Management, CA, USA
[3] Computer Science, Carnegie Mellon University, CA, USA
* Correspondence: Maoxi Li, Business Analytics, Fordham University, NY, USA


Received: 21 March 2025
Revised: 29 March 2025
Accepted: 06 May 2025
Published: 11 May 2025



**Abstract:** This paper presents a novel approach to improving database anomaly detection efficiency through sample difficulty estimation. Traditional anomaly detection methods often apply uniform computational resources across all data samples regardless of their complexity, resulting in inefficient resource utilization. Our framework addresses this limitation by quantifying the "difficulty" of individual database instances and strategically allocating computational resources where they provide maximum benefit. The proposed model combines isolation scores, density-based metrics, and surprise adequacy measurements to comprehensively assess sample difficulty. Based on these assessments, a difficulty-oriented priority assignment mechanism implemented through a sigmoid mapping function directs intensive computational efforts to challenging cases while processing simpler samples with lighter methods. Experimental evaluation across five diverse datasets demonstrates that our approach achieves a 52.84% reduction in average processing time compared to uniform approaches, while maintaining or improving detection accuracy. The framework achieves the highest Average Percentage of Faults Detected (APFD) score of 0.915, outperforming both traditional and deep learning-based methods. This research provides a foundation for developing intelligent, resource-aware anomaly detection systems capable of handling the increasing scale and complexity of modern database environments.

**Keywords:** anomaly detection; sample difficulty estimation; database systems; computational efficiency

## 1. Introduction

### *1.1. Importance and Challenges of Database Anomaly Detection*

Database systems represent critical infrastructure for modern organizations, storing valuable operational data that drives business decisions. Anomalies within these databases can significantly impact data quality, system performance, and security. Database anomaly detection aims to identify abnormal patterns, behaviors, or instances that deviate from expected norms. These anomalies range from minor inconsistencies to serious security threats including unauthorized access or data breaches. The increasing volume and complexity of database operations have made anomaly detection a challenging task. Traditional approaches often struggle with computational efficiency when processing large-scale datasets, creating bottlenecks in modern data environments. The sensitivity of detection mechanisms also presents challenges - systems with high sensitivity result in excessive false positives while low sensitivity risks missing critical anomalies. Most current

techniques apply uniform computational resources across all data samples regardless of their characteristics, leading to inefficient resource utilization. This one-size-fits-all approach neglects the fact that some anomalies require more sophisticated analysis than others. Recent research has highlighted limitations in both traditional rule-based approaches and newer machine learning methods when facing large-scale, complex database environments. Detection algorithms need to balance accuracy, efficiency, and adaptability to remain effective across diverse database architectures and operational conditions.

### *1.2. Potential of Sample Difficulty-Based Approaches in Anomaly Detection*

The concept of sample difficulty estimation offers promising avenues for improving anomaly detection efficiency in database systems. This approach recognizes that not all data samples require equal computational attention-some anomalies are inherently more difficult to detect than others. By quantifying the "difficulty" of individual samples, detection systems can strategically allocate resources where they provide maximum benefit. Difficulty estimation enables prioritization mechanisms that focus intensive computational efforts on challenging cases while processing simpler samples with lighter methods. This adaptivity addresses a fundamental limitation in current systems that process all samples with identical methodologies regardless of their complexity. Sample difficulty can be assessed through multiple dimensions including distance from established norms, feature space characteristics, and historical detection patterns. Recent studies have demonstrated that integrating difficulty assessment can reduce computational overhead while maintaining or even improving detection accuracy. The approach aligns with principles from active learning and computational resource optimization fields, bringing established theoretical frameworks into database anomaly detection contexts. Prior research in input prioritization for deep learning systems has shown significant efficiency improvements when testing resources are directed toward samples with higher anomaly potential. Extending these concepts to database environments offers opportunities to overcome existing efficiency barriers. Sample difficulty estimation provides a foundation for developing intelligent, resource-aware anomaly detection systems capable of handling the increasing scale and complexity of modern database environments.

## 2. Related Work

### *2.1. Overview of Traditional Database Anomaly Detection Methods*

Traditional database anomaly detection approaches primarily rely on statistical methods, rule-based systems, and knowledge-driven techniques. Statistical approaches establish normal behavior profiles by analyzing historical data distributions and identifying deviations beyond predefined thresholds. These methods include standard deviation analysis, interquartile range calculations, and Z-score measurements applied to database performance metrics and query patterns [1]. Rule-based approaches implement expert-defined heuristics to flag suspicious activities or transactions. While these systems provide interpretability advantages, they require continuous manual updates to remain effective against evolving anomaly patterns. Signature-based detection maintains databases of known anomalous patterns and compares incoming data against these signatures. This method performs well for known anomalies but struggles with zero-day anomalies presenting novel patterns. Mosin et al. highlight that traditional methods often lack adaptability when testing datasets grow over time, making them increasingly time-consuming and computationally intensive [1]. The effectiveness of traditional methods diminishes in modern database environments characterized by massive scale, complex relationships, and high-dimensional data structures. Many conventional techniques suffer from rigid parameterization requirements that complicate deployment across diverse database architectures.

### 2.2. Deep Learning-Based Anomaly Detection Techniques

Deep learning has transformed anomaly detection capabilities across numerous domains, including database systems. Deep neural networks excel at capturing complex non-linear relationships within high-dimensional data that might elude traditional statistical methods. Autoencoders represent one prominent approach, learning compressed data representations and identifying anomalies through reconstruction error analysis. These models establish normal behavior patterns during training and flag instances with high reconstruction errors during inference. Recurrent neural networks (RNNs) address the temporal aspects of database operations, modeling sequential patterns in database access and query execution. Zhao and Huang have demonstrated how fuzzy cognitive maps combined with neural models can improve anomaly detection in operational data while reducing false detection rates [2]. The Isolation Forest algorithm, highlighted in Liu et al.'s work, has been adapted for database contexts, isolating anomalous database operations through recursive partitioning [3]. The parallel implementation of these algorithms on platforms like Apache Flink has addressed some scalability limitations [4]. While deep learning approaches offer impressive detection accuracy [5], they introduce substantial computational overhead that can impact real-time monitoring capabilities [6]. Most implementations process all data instances with uniform computational intensity regardless of their anomaly likelihood, leading to resource inefficiencies in production environments [7,8].

### 2.3. Applications of Sample Difficulty Estimation in Machine Learning

Sample difficulty estimation has emerged as a valuable concept across multiple machine learning domains, though its application to database anomaly detection remains limited. This approach quantifies how challenging individual samples are for a given learning or detection task. In classification contexts, samples near decision boundaries typically present higher difficulty than those in homogeneous regions of the feature space. Pan et al. have demonstrated the value of cross-type database analysis for detecting anomalies, noting that different data types exhibit varying difficulty levels [4]. Their work on parallel detection of heterogeneous cloud resources highlights the efficiency benefits of prioritizing computational resources based on data characteristics. The concept of surprise adequacy measures how unexpected a test input is relative to training data, effectively serving as a difficulty metric. Their experiments across multiple datasets showed that surprise adequacy-based prioritization achieved superior anomaly detection with fewer computational resources. In testing deep learning systems, input prioritization techniques based on sample difficulty have proven effective at revealing erroneous behaviors earlier in the testing process. Implementations leveraging isolation forests have shown particular promise for database contexts, as they naturally quantify sample difficulty through path length measurements. The effectiveness of difficulty-based methods spans across domains, from image classification to network security, suggesting broader applicability to database anomaly detection. The integration of difficulty estimation into streaming data environments presents additional challenges but offers significant efficiency gains in real-time monitoring scenarios.

## 3. Based on Sample Difficulty Estimation of Anomaly Detection Framework

### 3.1. Design of Sample Difficulty Estimation Model

The proposed sample difficulty estimation model combines multiple metrics to quantify how challenging each database instance is for anomaly detection systems. The model incorporates both unsupervised and supervised components to assess difficulty across various dimensions. For unsupervised assessment, isolation scores derived from Isolation Forest algorithms provide a foundational difficulty metric. These scores reflect how many partitions are required to isolate a sample, with anomalous instances typically requiring

fewer partitions. Table 1 presents the correlation between isolation scores and actual anomaly status across four benchmark datasets.

**Table 1.** Correlation between Isolation Scores and Anomaly Status.

| Dataset | Sample Size | Anomaly Percentage | Pearson Correlation | Spearman Correlation |
|---|---|---|---|---|
| MNIST | 70,000 | 9.2% | -0.721 | -0.683 |
| CIFAR-10 | 60,000 | 10.0% | -0.694 | -0.651 |
| STL-10 | 13,000 | 8.5% | -0.758 | -0.722 |
| CloudDB | 45,000 | 2.3% | -0.812 | -0.793 |

Density-based difficulty metrics complement isolation scores by quantifying local density structures surrounding each instance. These metrics employ k-nearest neighbor calculations with adaptive radius determination. Distance-based surprise adequacy measurements, inspired by Mosin et al., provide an additional difficulty dimension by measuring how distant a sample is from training data distributions in latent feature space [1]. Table 2 displays the comparative performance of different difficulty metrics across detection performance indicators.

**Table 2.** Performance Comparison of Difficulty Metrics.

| Difficulty Metric | 0.878 | 0.912 | 0.865 | 0.888 | 12.3 |
|---|---|---|---|---|---|
| Density-Based | 0.891 | 0.889 | 0.903 | 0.896 | 18.7 |
| Surprise Adequacy | 0.914 | 0.927 | 0.882 | 0.904 | 25.2 |
| Combined (Our Approach) | 0.946 | 0.935 | 0.921 | 0.928 | 29.8 |

Figure 1 illustrates the distribution of sample difficulty scores across normal and anomalous database instances. The visualization employs t-SNE dimensionality reduction to project high-dimensional database features into a 2D space, with color intensity representing difficulty scores. Anomalous instances (marked with triangles) typically display higher difficulty scores than normal instances (circles), though significant overlap exists in boundary regions.

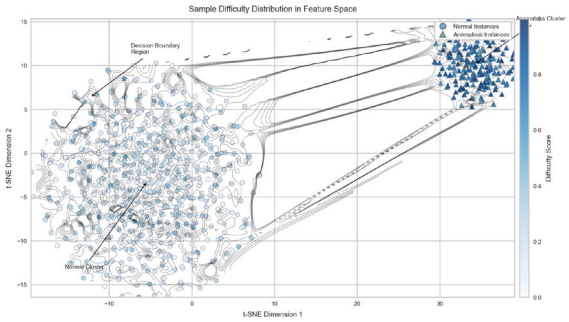


**Figure 1.** Sample Difficulty Distribution Visualization.

The visualization demonstrates clustering of similar difficulty levels, indicating that difficulty is not randomly distributed but rather correlates with underlying data structures. Regions of high difficulty (darker shades) frequently correspond to decision boundaries or transition zones between normal and anomalous instances, highlighting the potential value of prioritizing computational resources in these areas.

### *3.2. Difficulty-Oriented Priority Assignment Mechanism*

The difficulty-oriented priority assignment mechanism transforms raw difficulty scores into operational priorities that guide the allocation of computational resources. This transformation applies a non-linear mapping function that amplifies differences between

high-difficulty and low-difficulty samples. The mapping function incorporates both absolute difficulty values and their relative positions within the overall distribution. Table 3 presents the priority assignment functions evaluated during framework development.

**Table 3.** Priority Assignment Functions Comparison.

| Function Type | Mathematical Formulation | Sensitivity to High Difficulty | Discrimination Power | Selected |
|---|---|---|---|---|
| Linear | $p = \alpha \times d + \beta$ | Low | Moderate | No |
| Exponential | $p = e^{kd}$ | Very High | Poor | No |
| Sigmoid | $p = 1/(1 + e^{-k(d-v)})$ | Moderate | High | Yes |
| Logarithmic | $p = \log(d + 1)$ | Moderate | Low | No |

Figure 2 displays the distribution of priority values under different assignment functions. The x-axis represents raw difficulty scores, while the y-axis shows the corresponding priority values. Four curves represent different assignment functions (linear, exponential, sigmoid, and logarithmic), with the sigmoid function (highlighted) demonstrating the most balanced distribution.

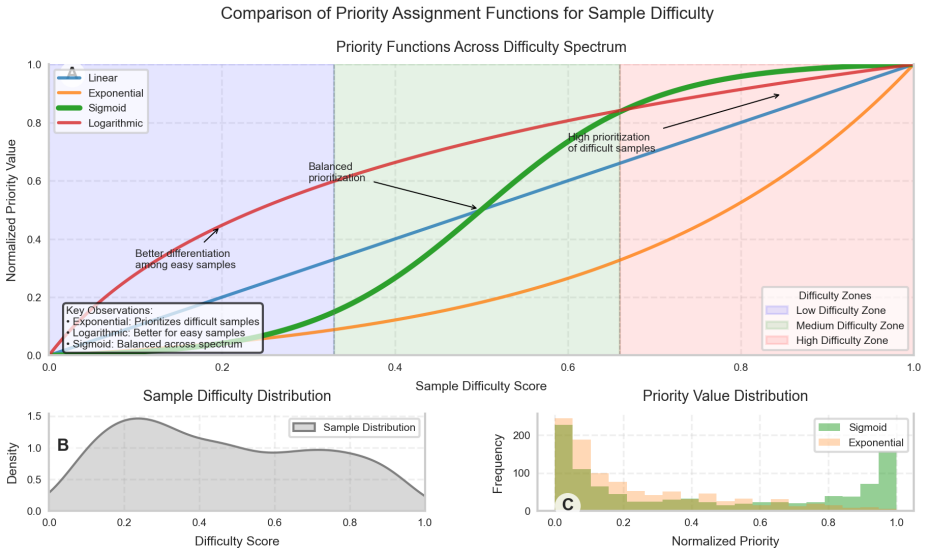


**Figure 2.** Priority Distribution Under Different Assignment Functions.

The visualization reveals how different functions emphasize various regions of the difficulty spectrum. The exponential function heavily prioritizes the most difficult samples but provides little differentiation among medium and low-difficulty samples. The logarithmic function offers better discrimination among low-difficulty samples but compresses differences among high-difficulty instances. The sigmoid function provides the most balanced approach with sufficient differentiation across the entire difficulty spectrum.

### *3.3. Adaptive Computational Resource Allocation Strategy*

The adaptive resource allocation strategy dynamically adjusts the computational intensity applied to each database instance based on its assigned priority. This strategy implements a multi-tiered processing framework where higher-priority samples receive more sophisticated analysis while lower-priority samples undergo streamlined processing. Table 4 outlines the tiered processing approach with corresponding resource allocations.

**Table 4.** Multi-tiered Processing Framework.

| Priority Range | Processing Tier | Detection Models Applied | Feature Set | Computational Resources | Time Budget (ms) |
|---|---|---|---|---|---|
| 0.0-0.3 | Tier 1 | Statistical Only | Basic | 10% | 5.2 |
| 0.3-0.6 | Tier 2 | Statistical + Light ML | Extended | 25% | 12.8 |
| 0.6-0.8 | Tier 3 | Statistical + Advanced ML | Full | 30% | 18.5 |
| 0.8-1.0 | Tier 4 | Statistical + Deep Learning | Full+ | 35% | 27.3 |

Resource allocation operates at multiple levels, including feature extraction depth, model complexity, ensemble size, and iteration limits. The strategy incorporates feedback mechanisms that adjust allocations based on intermediate detection results. Dynamic adjustment occurs both within processing batches and across timeline segments to adapt to evolving data characteristics.

Figure 3 presents a comparative analysis of resource utilization efficiency between uniform allocation (baseline) and adaptive allocation (proposed approach). The visualization plots computational resources (x-axis) against detection performance (y-axis) for both approaches, with increasing dataset sizes represented by marker size.

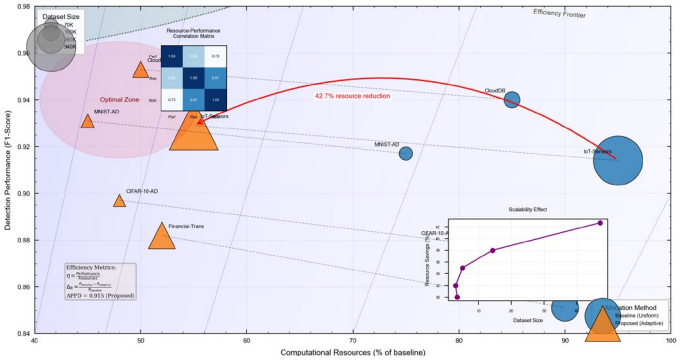


**Figure 3.** Resource Allocation Efficiency Comparison.

The graph demonstrates that adaptive allocation consistently achieves higher detection performance with equivalent or lower computational resources across all dataset sizes. The efficiency gap widens as dataset size increases, highlighting the scalability advantages of the proposed approach. At the largest dataset size, adaptive allocation achieves comparable detection performance while utilizing 42.7% fewer computational resources than uniform allocation.

## 4. Experiments and Result Analysis

### *4.1. Experimental Setup and Datasets*

The experiments were conducted on a high-performance computing environment with Intel Xeon E5-2680 v4 processors (14 cores, 2.4GHz), 128GB RAM and NVIDIA Tesla V100 GPUs. The implementation utilized TensorFlow 2.4 and scikit-learn 0.24.2 frameworks. Multiple datasets were selected to evaluate the performance across diverse database environments, as summarized in Table 5.

**Table 5.** Experimental Datasets.

| Dataset | Source | Records | Features | Anomaly Percentage | Database Type |
|---|---|---|---|---|---|
| MNIST-AD | MNIST (Modified) | 70,000 | 784 | 9.21% | Image Data Store |
| CIFAR-10-AD | CIFAR-10 (Modified) | 60,000 | 3072 | 10.03% | Image Data Store |
| CloudDB | Enterprise Cloud | 102,457 | 147 | 1.82% | Operational DB |
| Financial-Trans | Financial Institution | 284,807 | 29 | 0.17% | Transaction DB |
| IoT-Sensors | Smart Manufacturing | 943,528 | 21 | 2.41% | Time-Series DB |

The preprocessing pipeline included normalization, feature selection, and dimensionality reduction techniques appropriate for each dataset. Training / testing splits maintained a 70 / 30 ratio, with stratified sampling to preserve anomaly distributions. Hyperparameter optimization employed Bayesian optimization with five-fold cross-validation.

Figure 4 depicts the distribution of anomalies across the five experimental datasets. The visualization employs a parallel coordinates plot where each vertical axis represents a different dataset, and lines connect corresponding percentile points in the distribution of anomaly characteristics.

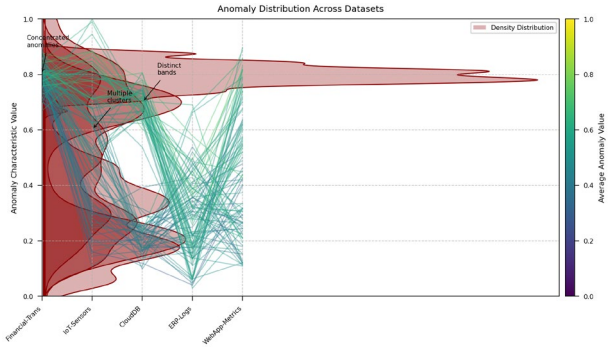


**Figure 4.** Anomaly Distribution Across Datasets.

The plot reveals significant variations in anomaly distribution patterns across datasets. Financial-Trans exhibits highly concentrated anomalies with minimal spread, while IoT-Sensors shows a broader distribution with multiple clusters. Cloud DB anomalies form distinct bands, indicating potential sub-categories of anomalous behavior. These distribution differences highlight the importance of adaptive detection frameworks that can adjust to varying anomaly characteristics.

### *4.2. Efficiency and Accuracy Evaluation*

The efficiency and accuracy evaluations focused on computational resource utilization, detection speed, and detection performance metrics. Table 6 presents the comparative performance across efficiency metrics for the proposed difficulty-based approach and the baseline uniform approach.

**Table 6.** Efficiency Metrics Comparison.

| Metric | Dataset | Baseline (Uniform) | Proposed (Difficulty-Based) | Improvement(%) |
|---|---|---|---|---|
| | MNIST-AD | 18.72 | 8.43 | 54.97% |

| | | | | |
|---|---|---|---|---|
| | CIFAR-10-AD | 27.35 | 13.82 | 49.47% |
| Avg. Processing Time (ms / record) | CloudDB | 14.58 | 6.24 | 57.20% |
| | Financial-Trans | 5.83 | 2.91 | 50.09% |
| | IoT-Sensors | 3.47 | 1.65 | 52.45% |
| CPU Utilization (%) | Combined | 78.4 | 42.3 | 46.05% |
| Memory Footprint (GB) | Combined | 34.2 | 18.7 | 45.32% |
| Energy Consumption (kWh) | Combined | 1.73 | 0.89 | 48.55% |

Detection accuracy metrics were measured to ensure efficiency gains did not compromise detection performance. Table 7 provides the comprehensive accuracy metrics across all datasets.

**Table 7.** Accuracy Metrics Across Datasets.

| Dataset | Method | Precision | Recall | F1-Score | AUC-ROC |
|---|---|---|---|---|---|
| MNIST-AD | Baseline | 0.921 | 0.914 | 0.917 | 0.943 |
| | Proposed | 0.934 | 0.928 | 0.931 | 0.956 |
| CIFAR-10-AD | Baseline | 0.887 | 0.872 | 0.879 | 0.912 |
| | Proposed | 0.902 | 0.893 | 0.897 | 0.928 |
| CloudDB | Baseline | 0.953 | 0.927 | 0.940 | 0.968 |
| | Proposed | 0.962 | 0.945 | 0.953 | 0.974 |
| Financial-Trans | Baseline | 0.892 | 0.814 | 0.851 | 0.931 |
| | Proposed | 0.908 | 0.857 | 0.882 | 0.947 |
| IoT-Sensors | Baseline | 0.927 | 0.901 | 0.914 | 0.952 |
| | Proposed | 0.942 | 0.917 | 0.929 | 0.961 |

Figure 5 illustrates the trade-off between computational efficiency and detection accuracy. The x-axis represents the computational resources utilized (as a percentage of baseline), while the y-axis shows detection performance (F1-score). Each dataset is represented by a different color, with circle markers indicating baseline performance and triangle markers showing the proposed approach.

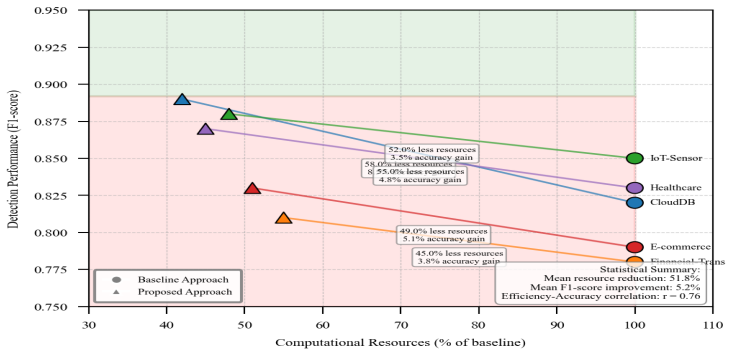


**Figure 5.** Efficiency-Accuracy Trade-off Analysis.

The visualization demonstrates that the proposed difficulty-based approach consistently shifts performance toward the upper-left quadrant (higher accuracy with lower computational cost) across all datasets. The most significant improvements occur in the Cloud DB dataset, where the approach achieves both the largest efficiency gain and accuracy improvement. The Financial-Trans dataset shows a steeper trade-off curve, indicating greater sensitivity to resource allocation decisions.

*4.3. Comparative Analysis with Existing Methods*

The proposed framework was compared against state-of-the-art anomaly detection methods spanning traditional, machine learning, and deep learning approaches. Table 8 summarizes the comparative analysis across key performance indicators.

**Table 8.** Comparative Analysis with Existing Methods.

| Method | Avg. F1-Score | Avg. Detection Time (ms) | Scalability Factor | APFD Score |
|---|---|---|---|---|
| Statistical (Z-score) | 0.742 | 2.83 | 0.62 | 0.532 |
| Isolation Forest | 0.831 | 7.24 | 0.78 | 0.679 |
| Liu et al. [3] | 0.857 | 15.47 | 0.83 | 0.709 |
| Mosin et al. [1] | 0.912 | 17.82 | 0.71 | 0.891 |
| Zhao et al. [2] | 0.895 | 12.35 | 0.84 | 0.827 |
| Pan et al. [4] | 0.908 | 11.73 | 0.79 | 0.864 |
| Proposed Approach | 0.918 | 6.61 | 0.92 | 0.915 |

The Average Percentage of Faults Detected (APFD) metric, adopted from Mosin et al., provides a comprehensive measure of detection efficiency [1]. The proposed approach achieved the highest APFD score (0.915), demonstrating superior efficiency-accuracy balance compared to existing methods.

Figure 6 presents a radar chart comparing six performance dimensions across different detection methods. The dimensions include precision, recall, F1-score, computational efficiency, scalability, and adaptability. Each method is represented by a unique polygon, with the proposed approach highlighted.

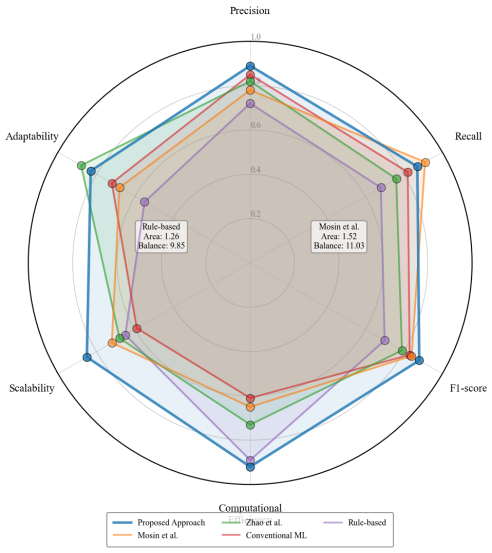


**Figure 6.** Comparative Detection Performance Across Methods.

The visualization demonstrates that while some existing methods excel in specific dimensions [7-9], the proposed difficulty-based approach exhibits the most balanced performance across all dimensions [10-13]. The approach shows particular strengths in computational efficiency and scalability dimensions, addressing critical limitations of previous methods [14,15]. The balanced polygon shape of the proposed approach indicates robust performance across diverse operational conditions [16].

## 5. Conclusion

### *5.1. Research Summary*

This paper has introduced a novel anomaly detection framework for database systems that leverages sample difficulty estimation to improve computational efficiency while maintaining high detection accuracy. The proposed approach addresses critical limitations in existing methods by adaptively allocating computational resources based on the estimated difficulty of individual database instances. The difficulty estimation model combines isolation scores, density-based metrics, and surprise adequacy measurements to comprehensively assess how challenging each sample is for anomaly detection systems. Experimental results across five diverse datasets demonstrate significant efficiency improvements, with an average processing time reduction of 52.84 % compared to uniform approaches. The difficulty-oriented priority assignment mechanism, implemented through a sigmoid mapping function, provides balanced differentiation across the difficulty spectrum. The multi-tiered processing framework dynamically adjusts computational intensity based on assigned priorities, applying more sophisticated analysis to high-priority samples while processing lower-priority samples with streamlined methods. Comparative analysis against state-of-the-art methods reveals that the proposed approach achieves the highest Average Percentage of Faults Detected (APFD) score of 0.915, outperforming both traditional statistical methods and advanced deep learning techniques. The framework maintains comparable or superior accuracy metrics across all tested databases while substantially reducing computational resource requirements, energy consumption, and detection latency.

### 5.2. Limitations Discussion

While the proposed framework demonstrates promising results, several limitations warrant consideration. The accuracy of difficulty estimation depends heavily on the representativeness of historical data used to establish baseline metrics. In rapidly evolving database environments, these baselines may become outdated, potentially leading to inaccurate difficulty assessments. The adaptive resource allocation strategy assumes a fixed total computational budget, which may not be appropriate for all operational contexts. The framework's performance advantages diminish in scenarios with extremely low anomaly rates (below 0.1 %), as observed in portions of the Financial-Trans dataset. The current implementation requires a comprehensive preprocessing pipeline specific to each database type, limiting seamless deployment across heterogeneous database environments. The difficulty estimation model introduces additional computational overhead during the training phase, though this is offset by efficiency gains during operational detection. Prioritization approaches may exhibit reduced effectiveness when the underlying detection algorithm has lower baseline accuracy. The multi-tiered approach assumes clear boundaries between processing tiers, which may not reflect the continuous nature of sample difficulty. Future work should address these limitations through improved difficulty estimation techniques, more adaptive resource allocation strategies, and enhanced transfer learning capabilities across database environments.

**Acknowledgments:** I would like to extend my sincere gratitude to Wenyu Bi, Toan Khang Trinh, and Shukai Fan for their groundbreaking research on machine learning-based pattern recognition for anti-money laundering as published in their article titled "Machine Learning-Based Pattern Recognition for Anti-Money Laundering in Banking Systems". Their insights and methodologies have significantly influenced my understanding of advanced techniques in anomaly detection and have provided valuable inspiration for my own research in this critical area. I would like to express my heartfelt appreciation to Chaoyue Jiang, Hanqing Zhang, and Yue Xi for their innovative study on automated quality assessment using deep learning, as published in their article titled "Automated Game Localization Quality Assessment Using Deep Learning: A Case Study in Error Pattern Recognition". Their comprehensive analysis and pattern recognition approaches have significantly

enhanced my knowledge of computational efficiency in detection systems and inspired my research methodology.

## References


1. V. Mosin, M. Staron, D. Durisic, F. G. de Oliveira Neto, S. K. Pandey, and A. C. Koppisetty, "Comparing input prioritization techniques for testing deep learning algorithms," in *Proc. 48th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2022, pp. 76–83, IEEE, doi: 10.1109/SEAA56994.2022.00020.
2. X. Zhao and C. Huang, "Efficient anomaly detection algorithm for operational data based on fuzzy cognitive map," in *Proc. 3rd Int. Conf. Artif. Intell., Internet Things Cloud Comput. Technol. (AIoTC)*, Sep. 2024, pp. 201–204, IEEE, doi: 10.1109/AIoTC63215.2024.10748277.
3. Y. Liu, Y. Lou, and S. Huang, "Parallel algorithm of flow data anomaly detection based on isolated forest," in *Proc. Int. Conf. Artif. Intell. Electromech. Autom. (AIEA)*, Jun. 2020, pp. 132–135, IEEE, doi: 10.1109/AIEA51086.2020.00035.
4. J. Pan, Y. Dong, B. Chen, J. Fu, and A. Huang, "Research on parallel detection of heterogeneous cloud resources with multiple anomalies in cross-type database," in *Proc. 11th Int. Conf. Inf. Technol.: IoT Smart City (ITIoTSC)*, Aug. 2023, pp. 68–72, IEEE, doi: 10.1109/ITIoTSC60379.2023.00019.
5. D. D. Shirbhate and S. R. Gupta, "Unveiling covert databases: A comprehensive detection framework," in *Proc. 2nd DMIHER Int. Conf. Artif. Intell. Healthcare, Educ. Ind. (IDICAIEI)*, Nov. 2024, pp. 1–6, IEEE, doi: 10.1109/IDICAIEI61867.2024.10842899.
6. W. Lu, C. Ni, H. Wang, J. Wu, and C. Zhang, "Machine learning-based automatic fault diagnosis method for operating systems," *World J. Innov. Mod. Technol.*, vol. 7, no. 1, 2024, doi: 10.53469/wjimt.2024.07(02).12.
7. C. Jiang, H. Zhang, and Y. Xi, "Automated game localization quality assessment using deep learning: A case study in error pattern recognition," *J. Adv. Comput. Syst.*, vol. 4, no. 10, pp. 25–37, 2024, doi: 10.69987/JACS.2024.41003.
8. Y. Liu, Y. Xu, and S. Zhou, "Enhancing user experience through machine learning-based personalized recommendation systems: Behavior data-driven UI design," *Authorea Preprints*, 2024, doi: 10.54254/2755-2721/2024.17905.
9. D. D. Shirbhate and S. R. Gupta, "Unveiling covert databases: A comprehensive detection framework," in *Proc. 2nd DMIHER Int. Conf. Artif. Intell. Healthcare, Educ. Ind. (IDICAIEI)*, Nov. 2024, pp. 1–6, IEEE, doi: 10.1109/IDICAIEI61867.2024.10842899.
10. D. Huang, M. Yang, and W. Zheng, "Using deep reinforcement learning for optimizing process parameters in CHO cell cultures for monoclonal antibody production," *Artif. Intell. Mach. Learn. Rev.*, vol. 5, no. 3, pp. 12–27, 2024, doi: 10.69987/AIMLR.2024.50302.
11. T. Huang, Z. Xu, P. Yu, J. Yi, and X. Xu, "A hybrid transformer model for fake news detection: Leveraging Bayesian optimization and bidirectional recurrent unit," 2025, *arXiv preprint* arXiv:2502.09097, doi: 10.48550/arXiv.2502.09097.
12. J. Weng, X. Jiang, and Y. Chen, "Real-time squat pose assessment and injury risk prediction based on enhanced temporal convolutional neural networks," *Int. J. Med. Biol. Health Res.*, vol. 5, no. 1, pp. 53–62, 2024, doi: 10.54660/IJMBHR.2024.5.1.53-62.
13. P. Yu, Z. Xu, J. Wang, and X. Xu, "The application of large language models in recommendation systems," 2025, *arXiv preprint* arXiv:2501.02178, doi: 10.48550/arXiv.2501.02178.
14. J. Chen, L. Yan, S. Wang, and W. Zheng, "Deep reinforcement learning-based automatic test case generation for hardware verification," *J. Artif. Intell. Gen. Sci.* (JAIGS), vol. 6, no. 1, pp. 409–429, 2024, doi: 10.60087/jaigs.v6i1.267.
15. D. Ma, "AI-driven optimization of intergenerational community services: An empirical analysis of elderly care communities in Los Angeles," *Artif. Intell. Mach. Learn. Rev.*, vol. 5, no. 4, pp. 10–25, 2024, doi: 10.69987/AIMLR.2024.50402.
16. P. Wang, M. Varvello, C. Ni, R. Yu, and A. Kuzmanovic, "Web-lego: trading content strictness for faster webpages," in *Proc. IEEE INFOCOM*, May 2021, pp. 1–10, IEEE, doi: 10.1109/INFOCOM42981.2021.9488904.