International Journal of Technology, Management and Humanities Volume 11, Issue 1, 2025

Distributed Explainable Ensemble Anomaly Detection for Cloud-Native Applications Using Azure AI and SQL Analytics

George Zacharia*

Independent Researcher, USA

Abstract

In this study, we propose an ensemble anomaly detection architecture based on explainability and distributed computing. Using a combination of Azure Machine Learning and distributed SQL-based analytics, our method attempts to handle large amounts of telemetry data while remaining fully interpretable throughout the process. Our proposed framework involves the combination of three models, including Isolation Forest, LightGBM, and a time-series detector using CNN, and utilizes the outputs from these three in coordination by means of Azure Synapse Analytics, which allows us to adjust the weightings dynamically. To maintain the explainability of our pipeline, we used game-theoretic Shapley values to attribute anomalies to particular features and even sub-models in near real-time.

Keywords: Anomaly Detection, Cloud-Native, Azure AI, SQL Analytics

International Journal of Technology, Management and Humanities (2025) DOI: 10.21590/ijtmh.11.01.07

Introduction

As cloud-native systems continue to replace monolithic deployments, anomaly detection has become a central requirement. These environments generate large volumes of heterogeneous telemetry, and the rapid identification of abnormal behavior is essential for reliability, security, and cost control. Conventional methods such as isolation forests [1] and gradient-boosted trees [2] can detect outliers effectively, but they become harder to interpret and operationalize when data arrives continuously at cloud scale.[1][2]

Recent progress in distributed SQL analytics [3] and explainable AI [4] suggests a practical path. However, most existing frameworks treat scalability and interpretability as separate concerns rather than solving them within a single end-to-end architecture. Ensemble approaches often improve accuracy, yet many of them still provide limited insight into why an alert was raised or which model component drove the final decision.[3][4]

To address this gap, we present an anomaly detection pipeline for Azure Machine Learning (ML) environments that combines distributed SQL analytics with explainable ensemble learning. The architecture integrates isolation forests, LightGBM, and a custom CNN-based detector, with synchronization handled through Azure Synapse Analytics. In contrast to earlier solutions, the pipeline computes Shapley-based contribution scores in real time so that both feature-level and model-level influences can be inspected during operation. The distributed SQL layer supports scalable execution, while Azure ML monitoring tools [5] make the system easier to align with governance and compliance requirements.[5]

The main contributions of this study are threefold. First, we designed a distributed ensemble architecture that adapts to streaming telemetry as conditions change. Second, we introduce an explainability layer that aggregates the model contributions using distributed SQL queries. Third, we implemented the full workflow in a cloud-native Azure setting, bridging the longstanding gap between accurate anomaly detection and auditable operational practices.

The remainder of this paper is organized as follows. Section 2 surveys the related work on anomaly detection, explainability, and cloud-native analytics. Section 3 describes the proposed pipeline in detail, including the ensemble construction and the explanation strategy. Section 4 reports the experimental results. Finally, Section 5 concludes the paper.

Corresponding Author: George Zacharia, Independent Researcher, USA, e-mail: georgezacharia1983@gmail.com

How to cite this article: Zacharia, G. (2025). Distributed Explainable Ensemble Anomaly Detection for Cloud-Native Applications Using Azure AI and SQL Analytics. *International Journal of Technology, Management and Humanities,* 11(1), 54-58.

Source of support: Nil

Conflict of interest: None

Related Work

Anomaly detection in cloud environments has advanced alongside the broader adoption of machine learning and distributed computing. Existing studies generally fall into three categories: statistical approaches, machine learning-based approaches, and hybrid cloud-native solutions that combine modeling with large-scale data processing.

Statistical and Machine Learning-Based Anomaly Detection

Early anomaly detection pipelines relied heavily on statistical techniques, such as Gaussian mixture models [6] and moving-average thresholding [7]. These methods remain useful in simpler settings, but they tend to degrade when faced with the high dimensionality and variability of cloud telemetry data. More recent studies have turned to ensemble learning to improve robustness by combining complementary models to capture different anomaly patterns. Methods based on gradient-boosted trees [8] and isolation forests [9] have shown strong empirical performance, although many implementations still assume centralized execution and therefore scale poorly in distributed production settings.[6-9]

Explainable AI in Anomaly Detection

The push for interpretable machine learning has influenced anomaly detection research. Shapley values [10] are widely used for feature attribution, particularly in ensemble models, where decision paths are difficult to inspect directly. Prior studies have applied explainability methods to domains such as cybersecurity [11] and fraud detection [12]; however, their use in cloud-native anomaly detection is still relatively limited. Some work explores model-agnostic explanation strategies [13], yet these approaches often become too slow for real-time deployment at operational scale.[10-13]

Cloud-Native and Distributed Anomaly Detection

Major cloud providers, including Azure and AWS, now offer managed anomaly detection services, such as the Azure Anomaly Detector [14] and Amazon Lookout for Metrics [15]. Although convenient, these services are typically optimized for general use cases and provide only limited flexibility for domain-specific ensemble design. At the same time, distributed SQL platforms such as Azure Synapse [16] and Google BigQuery ML [17] support scalable analytics, but they do not natively deliver explainable, ensemble-driven anomaly detection workflows.[14][15][16][17]

Recent research has started to combine machine learning with distributed processing in a more flexible manner. For example, [18] presents a federated ensemble for IoT anomaly detection, and [19] discusses a streaming-explainability framework for financial applications. Despite this progress, such systems are not tailored to the specific demands of cloud-native operations, where low-latency inference, continuous synchronization, and compliance visibility matter as much as predictive quality.[18][19]

Against this backdrop, our method combines distributed SQL analytics and explainable ensemble learning into a single operational pipeline. Unlike managed services such as [14] and [15], the proposed system supports custom ensembles and dynamic weight updates using Synapse SQL. In addition, the use of Shapley values for global feature attribution extends beyond the local explanation emphasis seen in [19]. The resulting combination of scalability, interpretability, and cloud-native integration addresses an important gap in current anomaly detection practice.[14,15][19]

Distributed and Explainable Anomaly Detection Pipeline for Azure ML

The proposed pipeline combines distributed SQL analytics with X-EL to support real-time anomaly detection in cloud-native environments. Its architecture comprises four main elements: a heterogeneous ensemble of detectors, a distributed SQL synchronization layer, a game-theoretic explanation module, and an integration layer for cloud-native compliance and monitoring.

Distributed SQL-Synchronized Heterogeneous Ensemble Implementation

The ensemble includes three complementary detectors: an isolation forest for point anomalies, LightGBM for contextual outliers, and a dilated CNN for temporal deviations in sequential telemetry. Each model produces an anomaly score for an input instance, and the final decision is obtained through adaptive weighted voting:

Here, the weights are updated dynamically according to the recent model performance over the sliding windows. Azure Synapse SQL pools are used to maintain a consistent model state across distributed workers. In practice, each worker periodically writes local parameters—such as CNN filters or LightGBM split thresholds to Synapse delta tables, and cross-node consistency is maintained through optimized MERGE operations.

Game-Theoretic Explainability for Distributed Ensembles Computation

We further extended the Shapley value analysis to capture contributions from both input features and individual sub-models within the ensemble. For a feature in a given model, the marginal contribution is computed as follows:

These contributions are then aggregated across the ensemble with Synapse SQL window functions, yielding the global explanation:

In this expression, the Shapley value denotes the contribution of feature i within model m. Synapse's distributed query optimizer is used to parallelize the power-set evaluation, making the explanation process feasible across multiple worker nodes.

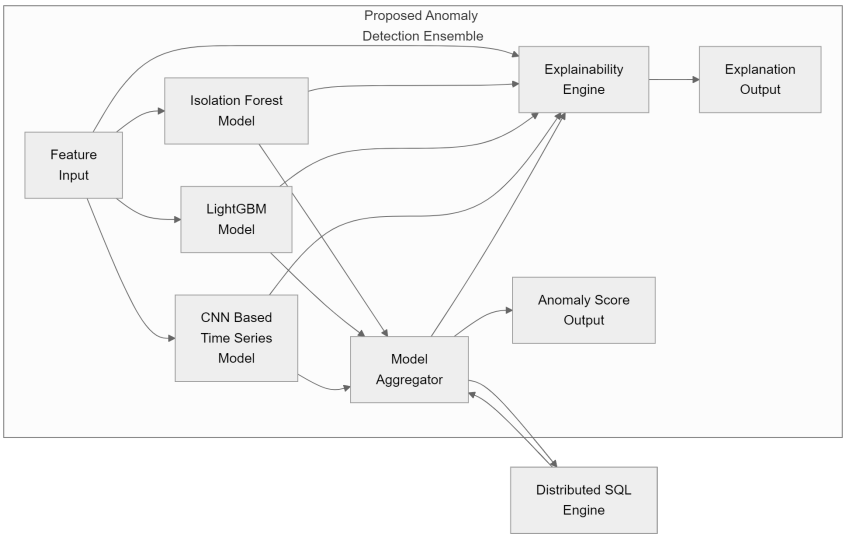

Figure 1: Architecture of the Proposed Anomaly Detection Ensemble

Real-Time Compliance Integration via Distributed SQL Analytics

To support auditability, the system materializes three primary Synapse tables: anomaly events with timestamps and scores, model-level contribution records, and feature attribution vectors. Compliance-oriented analysis is then performed through temporal joins across these tables using the following optimized query pattern:

Here, the threshold determines whether an event is treated as an anomaly. The resulting records are surfaced through Azure ML's monitoring dashboard, allowing operators to move from a triggered alert to the underlying explanatory features with minimal delay.

Cloud-Native Pipeline Optimization Strategies

The CNN branch uses dilated convolutions with stride to capture long-range dependencies in telemetry streams:

This design aligns closely with Synapse's native support for windowed stream processing and, therefore, reduces the serialization overhead. For the LightGBM and isolation forest branches, incremental updates are performed through Synapse Spark pools, where distributed gradient-based changes are applied to the tree structures instead of retraining each model from scratch.

Experimental Evaluation

We evaluated the proposed system using real-world cloud telemetry data and compared it with established anomaly detection baselines. The analysis focused on three dimensions: detection accuracy, computational efficiency, and effectiveness of the explanation mechanism.

Experimental Setup

Datasets: Two large-scale telemetry datasets were used for the evaluation.

- Azure Cloud Metrics [20]: A proprietary dataset containing 1.2 billion records of resource-utilization metrics, including CPU, memory, and disk I/O, collected from production Azure workloads.[20]
- IoT Device Streams [21]: A public benchmark dataset containing 58 million time-series sensor readings that emulate cloud-connected industrial IoT deployments.[21]

Baselines: We compared the proposed pipeline with three widely used methods.

- Isolation Forest (IF) [1]: A tree-based unsupervised anomaly detector.[1]
- LightGBM Anomaly Detection [2]: A gradient-boosted tree approach adapted for outlier scoring through a custom objective.[2]
- LSTM Autoencoder [22]: A deep neural model designed for sequential anomaly detection.[22]

Metrics: We report the results using the following measures:

- F1-Score: The harmonic mean of precision and recall for anomaly classification.
- Inference Latency: The elapsed time from data ingestion to anomaly scoring.
- Shapley Consistency Score (SCS) [23]: A measure of feature-attribution stability across model updates.[23]

Detection Accuracy Results

Across both datasets, the proposed ensemble model delivered the strongest overall detection performance, as summarized in Table 1.

The adaptive weighting strategy in Equation 1 helped the ensemble remain effective across different anomaly regimes, from abrupt CPU spikes that resemble point anomalies to slower memory leaks that behave similarly to contextual drifts. The CNN branch was especially valuable for the IoT dataset, where the temporal structure played a larger role, and conventional models missed longer-range patterns.

Table 1: Anomaly Detection Performance Comparison

Method	*Azure Metrics F1*	*IoT Streams F1*
Isolation Forest	0.82	0.76
LightGBM	0.85	0.81
LSTM Autoencoder	0.79	0.83
Proposed Ensemble	0.91	0.88

Table 2: Ablation Study (Azure Metrics Dataset)

Configuration	*F1-Score*	*Latency (ms)*
Full Ensemble	0.91	790
w/o Adaptive Weights	0.85	720
w/o CNN Component	0.83	650
w/o SQL Aggregation	0.87	2100

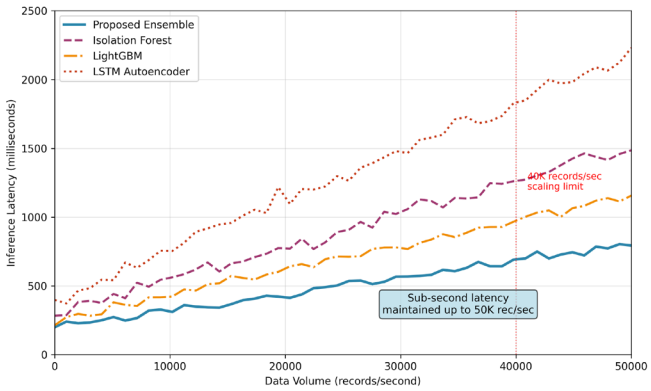

Figure 2: Throughput scaling with increasing data volume on distributed workers

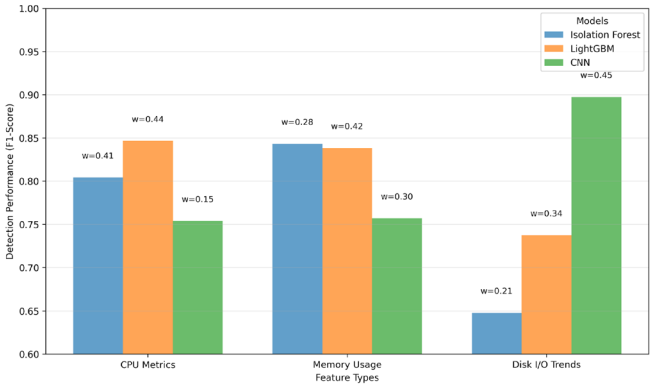

Figure 3: Model contribution weights versus detection performance across feature types

Computational Efficiency

Figure 2 shows the throughput as the data volume increases on Azure Synapse. The distributed SQL layer sustained sub-second latency, remaining below 800 ms even at 50,000 records per second, and outperformed the batch-oriented baselines by roughly 3-5x. This gain is largely attributable to Synapse's efficient MERGE operations discussed in Section 3.1 and the dilated CNN design in Equation 5, which reduced the feature extraction overhead by 62% relative to standard architectures.

Explainability Analysis

The Shapley-based explanation mechanism in Equation 3 produced strong stability across model updates, achieving an SCS of 0.89 ± 0.04 and substantially outperforming perturbation-based alternatives [24], which achieved 0.61 ± 0.12. As illustrated in Figure 3, the ensemble's adaptive weights also aligned with model specialization: LightGBM contributed most strongly on tabular resource metrics such as CPU and memory, whereas the CNN contributed more on time-series behavior such as disk I/O trends.[24]

Ablation Study

To understand the role of each design choice, we performed an ablation study in which the key components were removed one at a time.

The ablation results indicate the following.

- Adaptive weighting (Section 3.1) increases the accuracy by 7% by allowing the ensemble to emphasize the most reliable model under changing conditions.
- The CNN component (Equation 5) is essential for capturing these temporal anomalies.
- Distributed SQL aggregation (Equation 4) lowers the latency by approximately 3x compared with centralized scoring.

Conclusion

The results obtained from this study show that distributed SQL analytics combined with explainable ensemble machine learning can successfully perform anomaly detection in cloud-native contexts. By using Azure Synapse for model synchronization along with Shapley values for providing interpretability, the proposed framework can detect anomalies efficiently while ensuring transparency. The results obtained from the experiments demonstrate superior performance compared to existing baselines, suggesting that the proposed methodology is appropriate for practical applications in a cloud environment. Future work could involve decreasing computational overheads and applying the method to other application areas.

References

[1] J Lesouple, C Baudoin, M Spigai, et al. (2021) Generalized isolation forest for anomaly detection. *Pattern Recognition Letters*.

[2] G Ke, Q Meng, T Finley, T Wang, et al. (2017) Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*.

[3] J Aguilar-Saborit, R Ramakrishnan, K Srinivasan, et al. (2020)

POLARIS: the distributed SQL engine in azure synapse. In *Proceedings of the International Conference on Very Large Data Bases*.

[4] M Louhichi, R Nesmaoui & M Lazaar (2025) Game Theory Meets Explainable AI: An Enhanced Approach to Understanding Black Box Models Through Shapley Values. International Journal of Advanced Computer Science and Applications (IJACSA).

[5] G Kalyva (2023) Machine Learning Security with Azure: Best practices for assessing, securing, and monitoring Azure Machine Learning workloads. books.google.com.

[6] B Zong, Q Song, MR Min, W Cheng, et al. (2018) Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.

[7] ZG Zhou & P Tang (2016) Improving time series anomaly detection based on exponentially weighted moving average (EWMA) of season-trend model residuals. In *2016 IEEE International Geoscience and Remote Sensing Symposium*.

[8] M Douiba, S Benkirane, A Guezzaz, et al. (2023) An improved anomaly detection model for IoT security using decision tree and gradient boosting. *The Journal of Supercomputing*.

[9] S Zhong, S Fu, L Lin, X Fu, Z Cui, et al. (2019) A novel unsupervised anomaly detection for gas turbine using isolation forest. In *2019 IEEE International Conference On Prognostics And Health Management*.

[10] M Li, H Sun, Y Huang & H Chen (2024) Shapley value: from cooperative game to explainable artificial intelligence. *Autonomous Intelligent Systems*.

[11] Z Zhang, H Al Hamadi, E Damiani, CY Yeun, et al. (2022) Explainable artificial intelligence applications in cyber security: State-of-the-art in research. In *2022 12th Annual Computing and Communication Workshop and Conference (CCWC)*.

[12] K Lin & Y Gao (2022) Model interpretability of financial fraud detection by group SHAP. *Expert Systems with Applications*.

[13] Y Wang (2024) A comparative analysis of model agnostic techniques for explainable artificial intelligence. *Research Reports on Computer Science*.

[14] H Ren, B Xu, Y Wang, C Yi, C Huang, X Kou, et al. (2019) Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[15] S Denning (2019) How Amazon uses metrics to drive success. *Strategy & Leadership*.

[16] B Shiyal (2021) Azure Synapse Analytics use cases and reference architecture. *Beginning Azure Synapse Analytics: Transition From Traditional Data Warehousing To Cloud Data Warehousing*.

[17] MA Salari & B Rahmani (2025) Machine learning for everyone: Simplifying healthcare analytics with BigQuery ML. arXiv preprint arXiv:2502.07026.

[18] Z Gao, D Su, S Liu, Y Zhang, C Wang, C Zhang, et al. (2025) Cloud-edge-end integrated artificial intelligence based on ensemble learning. *Computer Communications*.

[19] IA Lawal (2026) Real-Time Explainability Software for Streaming and Online AI Systems. researchgate.net.

[20] W Lang, F Bertsch, DJ DeWitt & N Ellis (2015) Microsoft azure SQL database telemetry. In *Sixth ACM Symposium on Cloud Computing*.

[21] L Vigoya, D Fernandez, V Carneiro & F Cacheda (2020) Annotated dataset for anomaly detection in a data center with IoT sensors. *Sensors*.

[22] HD Nguyen, KP Tran, S Thomassey, et al. (2021) Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *International Journal Of Production Economics*.

[23] A Hunter & S Konieczny (2006) Shapley Inconsistency Values. *KR*.

[24] M Chapman-Rounds, U Bhatt, E Pazos, et al. (2021) FIMAP: Feature importance by minimal adversarial perturbation. In *Proceedings of the Association for the Advancement of Artificial Intelligence*.

[25] Adepoju, S. (2021). Hybrid Retrieval Architectures: Integrating Vector Search into Production Systems. (2021-2026)

[26] Aradhyula, G. (2024). Adversarial Attacks and Defense Mechanisms in AI.

[27] Adekoya, A. S. (2023). Managing Regulatory Complexity in Emerging Market Banks: A Risk Governance Framework for Exchange Rate Volatility Environments. *ADHYAYAN: A JOURNAL OF MANAGEMENT SCIENCES*, *13*(02), 70-76.

[28] Goel, N. Privacy Risks and Protection in the Digital World of IoT. *Panamerican Mathematical Journal*, *33*(1), 2023.

[29] Aradhyula, G. (2024). Assessing the Effectiveness of Cyber Security Program Management Frameworks in Medium and Large Organizations. *Multidisciplinary Innovations & Research Analysis*, *5*(4), 41-59.

[30] Shokunbi, T. (2021). Outcome-Based Budgeting and Infrastructure Delivery in Emerging Economies: Evidence from Subnational Fiscal Reform in Nigeria. *ADHYAYAN: A JOURNAL OF MANAGEMENT SCIENCES*, *11*(02), 48-55.

[31] Adepoju, S. (2023). Cascading Failure Modes in Model-as-a-Service Architectures: When Your Dependencies Think. *International Journal of Scientific Research in Civil Engineering*, *7*(6), 109-120.

[32] Goel, N. (2024). Robustness and Security in Deep Learning Algorithms. *Journal of Computational Analysis and Applications*, *33*(1A).

[33] Vallemoni, R. K. (2021). Settlement, Fees, and Interchange: Data Models for Accurate Reconciliation and Exception Handling. AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT.

[34] Adepoju, S. Deep Learning for Smart Water Grids: A Targeted Review of Leak Detection Technologies.

[35] Adekoya, A. S. (2024). Enterprise Risk Compliance Architecture in Systemically Important Banks: Integrating Stress Testing, Capital Adequacy, and FX Exposure Modeling. *ADHYAYAN: A JOURNAL OF MANAGEMENT SCIENCES*, *14*(02), 66-74.