

A Minimax-Based Context-Aware Sensor-Cloud Architecture with Integrated Machine Learning Analytics and Dynamic Model Orchestration

M. Jayalakshmi^{1,2}, Ts. Dr. Tadiwa Elisha Nyamasvisva³, Abudhahir Buhari³, K. Maharajan⁴

¹Post-Doctoral Researcher, Kuala Lumpur University of Science and Technology (KLUST), Kajang, Selangor, Malaysia

²Professor, Department of Computer Science and Engineering-Cyber Security, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India

³ Department of Computing, Faculty of Engineering, Science and Technology (FEST), Kuala Lumpur University of Science and Technology (KLUST), Kajang, Selangor, Malaysia

⁴Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Srivilliputhur, Tamil Nadu, India

*Corresponding Author: e-mail: jayalacsmi@gmail.com, ORCID: 0000-0002-7297-9161

Abstract

Internet of Things (IoT) sensor-cloud systems require reliable anomaly detection methods that account for changing environmental and operational conditions. Conventional threshold-based detectors and context-independent machine learning models may fail to identify anomalies whose significance depends on the operating context. This study presents a minimax-based extension of the Context-Aware Sensor-Cloud Architecture with Integrated Machine Learning Analytics and Dynamic Model Orchestration (CASC-MO). The architecture integrates context-specific Isolation Forest models, categorical context descriptors, contamination calibration, and a global fallback model. The proposed minimax selection strategy aims to select a candidate model by minimizing its maximum validation loss across the evaluated operational contexts. The framework is investigated using a simulated smart-building dataset containing 4,320 observations from five sensor channels collected over 30 days. The dataset includes point, contextual, collective, and sensor-drift anomalies. Experimental comparisons are designed to evaluate minimax-based selection against the original Hamming-distance-based CASC-MO method, a static-threshold detector, and a context-independent Isolation Forest. Evaluation metrics include precision, recall, F1-score, false alarm rate, area under the receiver operating characteristic curve (AUC-ROC), and inference latency. The study also examines performance across anomaly categories and the limitations of using simulated data. The results are intended to establish whether minimax-based orchestration improves the reliability of anomaly detection across different operating contexts while maintaining acceptable computational cost.

Keywords: Anomaly detection, Context awareness, Isolation forest, Sensor-cloud architecture, Smart buildings

1. Introduction

The increasing use of Internet of Things (IoT) technologies has enabled continuous monitoring of smart buildings, industrial infrastructure, and other sensor-enabled environments. These systems generate multivariate data streams that require reliable processing, storage, and analysis to identify abnormal operating conditions. Sensor-cloud architectures support these requirements by connecting distributed sensing devices with cloud-based data management and analytical services. However, the effectiveness of such systems depends not only on collecting sensor measurements but also on interpreting them in relation to changing operational conditions. Research on IoT systems and machine learning demonstrates the importance of integrating sensing, communication, data processing, and intelligent analytics to support dependable monitoring (Alsharif *et al.*, 2020; Fortino *et al.*, 2020; Liu *et al.*, 2018). Conventional monitoring systems frequently rely on fixed thresholds to identify abnormal sensor readings. Although threshold-based approaches are computationally inexpensive and straightforward to implement, they may fail when acceptable sensor values vary according to occupancy, time of day, equipment operating mode, or other contextual conditions. For example, a temperature reading may be normal during occupied hours but indicate unnecessary heating during an unoccupied period. Contextual anomalies of this kind cannot always be identified by examining measurements independently of their operating conditions. Studies on occupancy prediction and time-series anomaly

detection highlight the importance of analysing sensor behaviour in relation to the environment in which measurements are generated (Arief-Ang *et al.*, 2018; Cook *et al.*, 2019; Nassif *et al.*, 2021).

Machine learning provides alternative approaches to detecting unusual patterns in sensor data. Isolation Forest identifies observations that are isolated relatively quickly through random partitioning, making it suitable for computationally efficient anomaly detection. Extended Isolation Forest improves the treatment of decision boundaries, while robust random cut forests provide mechanisms for detecting anomalies in streaming data (Guha *et al.*, 2016; Hariri *et al.*, 2021). Other approaches, including sparse neural networks, variational autoencoders, and deep autoencoder architectures, have also been investigated for time-series and IoT anomaly detection (Gugulothu *et al.*, 2018; Kim *et al.*, 2019; Meidan *et al.*, 2018). Nevertheless, applying a single model to observations from multiple operating conditions can obscure differences between normal and abnormal behaviour within individual contexts.

Context-aware computing offers a complementary approach by representing environmental and operational conditions as structured information. Such context can support more appropriate interpretation of sensor observations and improve the relevance of analytical decisions. Meanwhile, stream-processing platforms such as Apache Flink demonstrate how continuous data streams can be processed efficiently, and research on cognitive edge computing highlights the potential for distributing intelligence closer to data sources (Carbone *et al.*, 2015; Cicirelli *et al.*, 2019). In network and service management, machine learning and zero-touch management research further emphasise automated analysis, adaptive decision-making, and closed-loop operation (Benzaid and Taleb, 2020; Boutaba *et al.*, 2018; de Sousa *et al.*, 2021). However, connecting context representation directly to runtime model selection remains an important architectural challenge.

A further consideration is the management of machine learning models throughout their operational lifecycle. Changes in context, data distributions, and system requirements can affect model reliability, while technical debt and observability limitations can complicate the maintenance of deployed analytical systems (Bhatia *et al.*, 2026; Dromard *et al.*, 2016; Murugesan and Ganesan, 2026). These concerns motivate architectures that can identify the current operating context, select a suitable model, and request adaptation when an unfamiliar context is encountered. The related context-aware sensor-cloud research provides a foundation for investigating this integration (Jayalakshmi *et al.*, 2026).

To address this challenge, this study presents the Context-Aware Sensor-Cloud Architecture with dynamic Model Orchestration (CASC-MO). The proposed five-layer architecture integrates sensing and data acquisition, context characterizations, model registration and orchestration, machine learning analytics, and cloud-based storage and feedback. Its orchestration mechanism represents operating conditions through structured context descriptors and uses Hamming distance to identify the closest registered context when an exact match is unavailable. Isolation Forest models are trained for sufficiently represented contexts, with contamination parameters calibrated according to the observed anomaly prevalence. A global fallback model supports observations associated with previously unregistered contexts.

The architecture is evaluated using a simulated smart-building dataset containing 4,320 observations across five sensor channels over 30 days. Four anomaly categories point, contextual, collective, and sensor drift are considered. The proposed approach is compared with a static-threshold detector and a context-agnostic Isolation Forest using precision, recall, F1-score, false alarm rate, AUC-ROC, and inference latency. In the reported experiment, tuned CASC-MO achieves an AUC-ROC of 0.882 and recall of 0.828, while its contextual-anomaly F1-score is 0.252; both baselines achieve zero F1-score for this category. However, none of the evaluated methods detects the injected sensor drift, and the higher recall of CASC-MO is accompanied by a higher false alarm rate. These findings establish the scope and limitations of the proposed architecture and motivate further validation on real-world datasets.

Despite advances in context-aware anomaly detection, an important research gap remains in selecting machine learning models that perform reliably across different operating contexts. Hamming-distance-based matching identifies the closest registered context, but it does not explicitly account for the worst-case detection loss across contexts. Consequently, the selected model may perform well in some operating conditions while producing higher detection errors in others. A minimax-based model selection strategy could address this limitation by selecting a model that minimizes the maximum validation loss across the considered contexts. However, its effectiveness must be established through experimental comparison with conventional nearest-context selection. This study therefore investigates the integration of minimax-based model selection into the CASC-MO architecture, with particular attention to overall anomaly detection performance, contextual anomalies, false-alarm rates, and inference latency.

The main objective of this study is to extend context-aware sensor-cloud model orchestration through minimax-based model selection. Unlike a strategy that selects a model only according to the nearest operational context, the proposed approach evaluates candidate models according to their worst-case validation loss across the represented contexts. The study compares the minimax-based strategy with the original nearest-context strategy to determine its effects on anomaly-detection performance and inference latency.

2. Materials and Methods

2.1 Study Design

This study adopts a design science approach to develop and evaluate the Context-Aware Sensor-Cloud Architecture with dynamic Model Orchestration (CASC-MO). The research focuses on integrating operational context with

machine learning-based anomaly detection in a simulated smart-building environment. The methodology comprises three main stages: architectural design, framework implementation, and experimental evaluation. During the design stage, the context characterizations and model orchestration components are defined. The implementation stage establishes context-specific anomaly detection models and a global fallback model. Finally, the experimental stage compares the proposed framework with two baseline methods to assess detection performance, contextual anomaly identification, and inference latency.

2.2 Dataset and Data Collection

The evaluation uses a simulated commercial smart-building dataset generated through physics-based simulation. It contains 4,320 observations collected at ten-minute intervals over 30 days, from 15 January to 13 February 2026. Five sensor channels are considered: temperature, relative humidity, illuminance, carbon dioxide concentration, and electrical power consumption. The simulation represents building conditions associated with occupancy, heating, ventilation, and lighting. Four anomaly categories are incorporated into the dataset: point anomalies, contextual anomalies, collective anomalies, and sensor drift. The dataset is divided chronologically into 3,024 training observations and 1,296 testing observations to preserve temporal ordering.

2.3 Context Characterisation

The context characterisation component identifies the operating conditions associated with each sensor observation. Four contextual attributes are used: time of day, day type, season, and equipment operating mode. Time of day includes categories such as night, morning, afternoon, and evening, while day type distinguishes weekdays from weekends. Seasonal information and operating modes provide additional details about building conditions. The attributes are encoded using categorical encoders and combined into a composite key, which is mapped to a context identifier. Sixteen context states are represented in the experimental evaluation. This representation enables the framework to associate sensor readings with their relevant operating conditions.

$$M^* = \arg \min_{M_j \in R} \max_{C_i \in \mathcal{C}} L(M_j, C_i)$$

Here, M^* denotes the selected model, R represents the registered candidate models, $\mathcal{C}$ denotes the set of operating contexts, and $L(M_j, C_i)$ represents the detection loss of model M_j under context C_i .

2.4 Model Training and Registry Construction

The model registry contains context-specific Isolation Forest models trained on observations belonging to individual context states. A minimum of 15 observations is required to train a separate model for a context. Each context-specific model uses 150 estimators, while the global fallback model uses 200 estimators. The maximum number of samples per tree is limited to the smaller of 100 and the available context sample count. A fixed random state of 42 supports reproducibility. Context-specific models are calibrated using the observed anomaly rate within their respective training subsets. The global model provides an alternative when a suitable context-specific model is unavailable.

2.5 Contamination Calibration

The framework adjusts model sensitivity according to the anomaly prevalence observed in each context. The empirical anomaly rate is calculated by dividing the number of labelled anomalous training observations by the total observations within that context. A calibration buffer of 0.02 is added to this rate, subject to a minimum contamination bound of 0.03 and a maximum bound of 0.12. This procedure prevents the contamination parameter from falling outside the specified range. Consequently, contexts with different anomaly distributions can use different calibration settings rather than relying on a single value for all operating conditions. The global fallback model supports observations associated with previously unregistered contexts.

2.6 Dynamic Model Orchestration

The orchestration engine selects an appropriate model for each incoming sensor observation using its context descriptor. When an exact context match is available, the corresponding context-specific model is selected. When an exact match is unavailable, Hamming distance is used to identify eligible candidate models from related contexts. The proposed minimax strategy then selects the candidate model that minimises the maximum validation loss across the evaluated operational contexts. If no eligible context-specific model is available, the global fallback model is used. The selected model applies its validated decision threshold to classify the observation as normal or anomalous.

2.6.1 Minimax-Based Contextual Model Selection

To incorporate minimax decision-making into the CASC-MO architecture, the model-orchestration component selects a registered model by minimising its worst-case validation loss across the relevant operational contexts. Let M denote the set of eligible models in the model registry and C_v denote the set of contexts represented in the validation data.

For a candidate model M_j and context C_i let $L(M_j, C_i)$ represent the validation loss, defined as $1 - F1(M_j, C_i)$, where $F1(M_j, C_i)$ is the F1-score of model (M_j) in context (C_i). The minimax selection rule is expressed as

$$M^* = \arg \min_{M_j \in \mathcal{M}} \max_{C_i \in \mathcal{C}_V} L(M_j, C_i)$$

$$L(M_j, C_i) = 1 - F1(M_j, C_i)$$

The minimax strategy selects the candidate model with the lowest worst-case validation loss across the represented operational contexts. The eligible candidate models, validation contexts, loss calculation, and fallback procedure are defined before evaluation. The selected model is then evaluated on the independent test set, which is not used for model selection or threshold tuning. The implementation and experimental results are reported below.

CASC-MO Inference Phase

Input: Observation s_t , context C_t , registry R , encoder E , threshold θ

Output: Binary anomaly decision $\hat{y}_t \in \{0, 1\}$

- 1: Encode $C_t \rightarrow \text{context_id}$ via E and composite key mapping
- 2: if $\text{context_id} \in R$ then
- 3: ($\text{scaler}, M_{\text{active}}$) $\leftarrow R[\text{context_id}]$ {exact match}
- 4: else
- 5: $j^* \leftarrow \text{argmin}_{\{j \in R\}} d_H(C_t, C_j)$ — Eq. (6)
- 6: ($\text{scaler}, M_{\text{active}}$) $\leftarrow R[j^*]$ {nearest neighbour}
- 7: if $d_H(C_t, C_{j^*}) \geq 2$ then log adaptation_request(context_id)
- 8: end if
- 9: $s_t_{\text{scaled}} \leftarrow \text{scaler.transform}(s_t)$
- 10: $\text{score} \leftarrow -M_{\text{active}}.\text{score_samples}(s_t_{\text{scaled}})$ — Eq. (7)
- 11: $\hat{y}_t \leftarrow \mathbb{1}[\text{score} > \theta]$
- 12: return $\hat{y}_t$

2.7 Anomaly Scoring and Threshold Selection

After model selection, the active Isolation Forest generates an anomaly score for each sensor observation using the negated output of the score samples method, with higher scores indicating greater anomalousness. A decision threshold converts the continuous anomaly score into a binary classification of normal or anomalous behaviour. Candidate thresholds are evaluated over percentiles ranging from (P_{70}) to (P_{97}), and the threshold that achieves the highest F1-score on the validation set is selected. The selected threshold is then fixed before evaluating the final model on the independent test set. The test set is not used for threshold selection or model tuning, ensuring that the reported performance reflects evaluation on previously unseen observations.

2.8 Baseline Methods and Experimental Comparison

The evaluation compares four configurations: a static-threshold detector, a context-agnostic Isolation Forest, the original Hamming-distance-based CASC-MO, and the minimax-based CASC-MO. The static-threshold detector applies predefined global limits, whereas the context-agnostic Isolation Forest models the aggregated sensor distribution without context-specific model selection. The original CASC-MO uses Hamming distance to select a registered model when an exact context match is unavailable. The minimax-based CASC-MO uses the minimax criterion to select among eligible candidate models by minimizing the maximum validation loss across the evaluated operational contexts. All configurations are assessed using the same data partitions and evaluation metrics. The comparison considers overall detection performance, performance across individual anomaly categories, false-alarm rate, and inference latency.

2.9 Evaluation Metrics

The evaluation uses precision, recall, F1-score, false alarm rate (FAR), area under the receiver operating characteristic curve (AUC-ROC), and inference latency. Precision measures the proportion of predicted anomalies that are genuine, whereas recall measures the proportion of actual anomalies detected. The F1-score combines precision and recall into a single measure. FAR represents the proportion of normal observations incorrectly classified as anomalous. AUC-ROC assesses the ability of continuous anomaly scores to distinguish anomalous observations from normal ones across decision thresholds. Inference latency measures the average time required to process an observation and produce a decision. These metrics assess detection effectiveness and computational cost.

2.10 Implementation Environment and Data Partitioning

The prototype was implemented using Python 3.12, NumPy 2.4, pandas 2.2, and scikit-learn 1.5. The experiments were conducted on a single Intel Xeon E5-2690 CPU core operating at 2.9 GHz without GPU acceleration. The dataset was partitioned chronologically, with 70% of the observations allocated to training and 30% to testing. This approach preserves the temporal sequence of observations rather than randomly mixing measurements from different periods. The training phase constructs the context-specific model registry and global fallback model, while the inference phase processes the test observations. The study reports average CASC-MO inference latency of approximately 17.23 milliseconds per observation.

3. Results

3.1 Dataset Characteristics and Experimental Setup

The dataset represents a simulated commercial smart-building environment monitored through five sensor channels. It contains 4,320 observations collected at ten-minute intervals over 30 days. The time-ordered partition provides 3,024 training observations and 1,296 testing observations as shown in Table 1. 16 context states represent different operating conditions, while four anomaly categories support the evaluation of detection capabilities. These characteristics establish the experimental basis for assessing context-aware anomaly detection and dynamic model orchestration.

Table 1. Dataset Characteristics and Experimental Setup

Parameter	Description / Value
Application domain	Commercial smart-building monitoring
Dataset generation method	Physics-based simulation
Dataset duration	30 days
Total number of observations	4,320
Number of sensor channels	5
Sensor measurements	Temperature, relative humidity, illuminance, CO ₂ concentration, electrical power
Sampling interval	10 minutes
Data collection period	15 January–13 February 2026
Training–testing split	70%–30%
Training observations	3,024
Testing observations	1,296
Context states	16
Anomaly categories	Point, contextual, collective, sensor drift

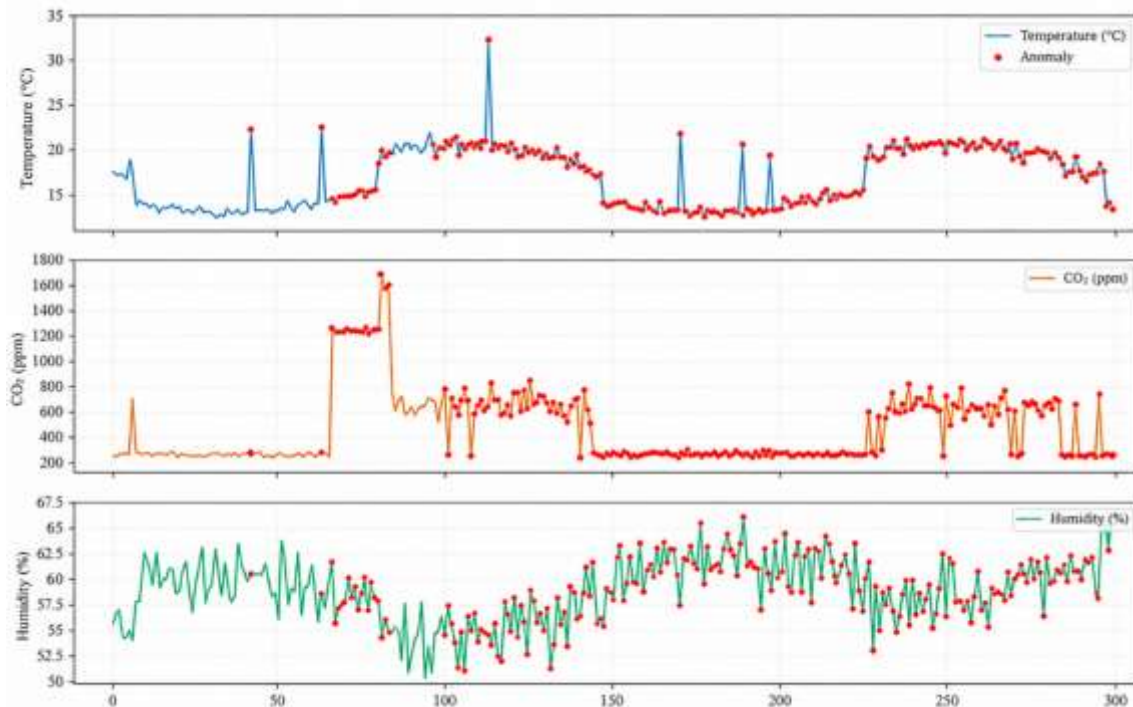


Figure 1. Sensor Readings and Anomaly Distribution

The figure 1 illustrates temperature, carbon dioxide (CO₂), and humidity measurements across a sequence of sensor observations. Red markers identify readings classified as anomalies, while the coloured lines represent the corresponding sensor measurements. Temperature anomalies include sudden increases and unusual fluctuations relative to surrounding values. The CO₂ plot shows substantial changes in concentration, including elevated readings and intermittent spikes. Humidity measurements exhibit variations across the observation period, with several points marked as anomalous. These patterns demonstrate that unusual sensor behaviour can appear differently across environmental variables, highlighting the importance of analysing multiple sensor channels when developing context-aware anomaly detection systems.

3.2 Hyperparameters and Configuration of the CASC-MO Framework

The framework configuration defines the parameters used to train context-specific Isolation Forest models and a global fallback model. Contamination bounds and a calibration buffer adjust model sensitivity to the anomaly rate within each context as shown in Table 2. Hamming-distance matching supports the selection of an appropriate registered model when an exact context match is unavailable. The adaptation trigger and validation-based threshold selection further control model updates and anomaly decisions. Together, these settings describe the operational configuration evaluated in the study.

Table 2. Hyperparameters and Configuration of the CASC-MO Framework

Component	Parameter	Value	Purpose
Context-specific Isolation Forest	Number of estimators	150	Balances variance and computational cost
Global fallback Isolation Forest	Number of estimators	200	Provides broader distributional coverage
Isolation Forest	Maximum samples	$\min(100, X_i)$	Limits samples used for each context model
Isolation Forest	Random state	42	Supports reproducibility
Contamination calibration	Minimum bound $\epsilon_{\min}$	0.03	Defines the lower calibration bound
Contamination calibration	Maximum bound $\epsilon_{\max}$	0.12	Defines the upper calibration bound
Contamination calibration	Buffer, δ	0.02	Provides a calibration buffer
Context matching	Hamming-distance threshold	1	Controls context-match acceptance
Model adaptation	Adaptation trigger θ_{adapt}	2	Defines the adaptation trigger
Decision threshold	Percentile range	$P_{70}\text{--}P_{97}$	Defines the threshold-selection range
Context-specific training	Minimum samples	15	Sets the minimum context sample count
Data partitioning	Training–testing split	70%–30%	Uses time-ordered partitioning

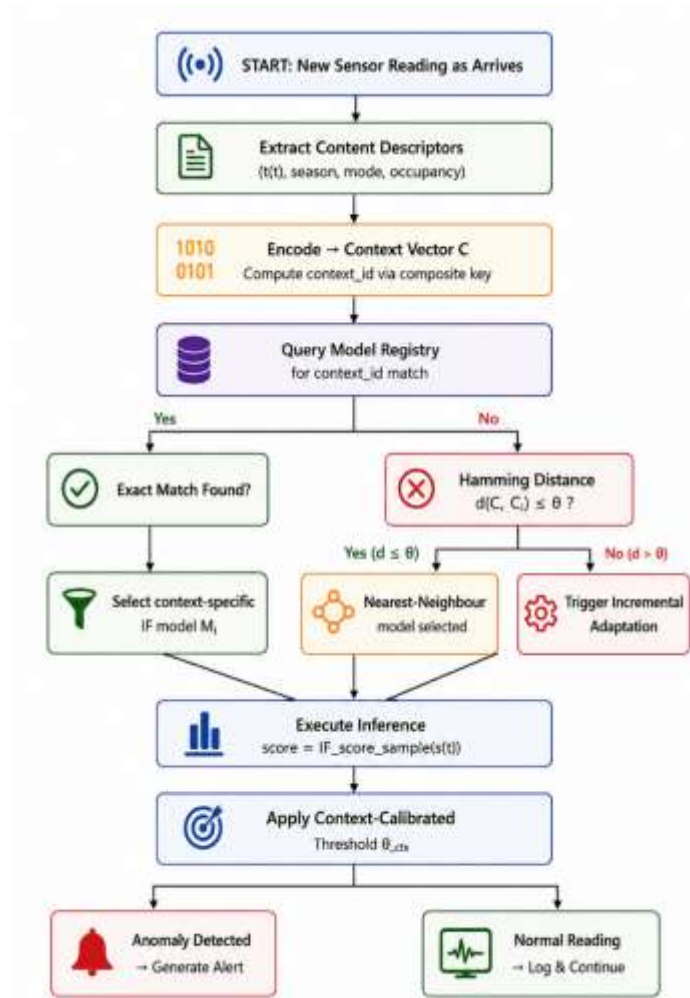


Figure 2. Workflow of the Proposed CASC-MO Architecture

The figure 2 presents the operational workflow of the proposed CASC-MO architecture. Processing begins when a new sensor reading arrives, followed by the extraction of contextual descriptors such as time, season, operating mode, and occupancy. These descriptors are encoded into a context vector that is used to query the model registry. If an exact match is unavailable, the architecture evaluates context similarity using Hamming distance to select a suitable neighbouring model or initiate incremental adaptation. The selected model performs inference, and a context-calibrated threshold determines whether the reading is anomalous. The final decision generates an alert or records the reading as normal.

3.3 Detection Performance on the Test Set

The overall evaluation compares CASC-MO with a static-threshold detector and a context-agnostic Isolation Forest. The comparison assesses precision, recall, F1-score, false-alarm rate, AUC-ROC, and inference latency on the test set.

Table 3. Detection Performance on the 1,296-Sample Test Set (281 Anomaly Samples, 1,015 Normal Samples)

Method	Precision	Recall	F1-score	FAR	AUC-ROC	Latency (ms)
Static Threshold (M1)	1.0000	0.4375	0.6087	0.0000	0.7188	< 0.001
Context-Agnostic IF (M2)	0.5600	0.4375	0.4912	0.0179	0.7098	0.017
CASC-MO —raw (M3)	0.4034	0.7500	0.5246	0.0576	0.8462	17.23
CASC-MO —tuned (M3+)	0.4046	0.8281	0.5436	0.0633	0.8824	17.23

Note: IF = Isolation Forest; FAR = false-alarm rate; AUC-ROC = area under the receiver operating characteristic curve. M3+ represents CASC-MO after validation-based threshold tuning, as described in the author's article.

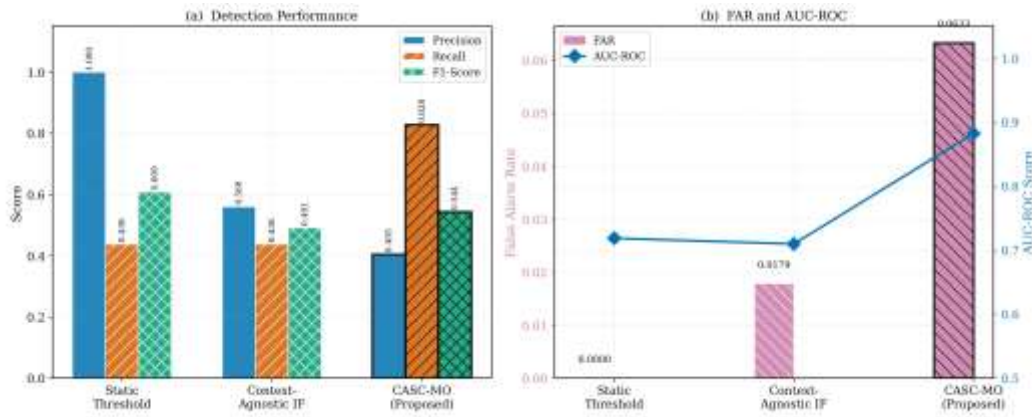


Figure 3. Detection Performance and False-Alarm Comparison

The figure 3 compares the proposed CASC-MO method with static-threshold and context-agnostic Isolation Forest approaches. CASC-MO achieves higher recall and AUC-ROC, indicating improved identification of anomalous observations and stronger overall discrimination between normal and abnormal readings. However, its precision is lower than that of the static-threshold method, and its false-alarm rate is higher than those of both comparison methods. The static-threshold approach achieves perfect precision in this evaluation but identifies fewer anomalies. These results highlight a trade-off between detecting more unusual observations and limiting false alarms. Therefore, method selection should consider the operational requirements of the intended monitoring environment.

3.4 F1-Score Comparison Across Anomaly Categories

The category-level evaluation examines how each method responds to point, contextual, collective, and sensor-drift anomalies. The tuned CASC-MO model achieves an F1-score of 0.2523 for contextual anomalies, whereas both baseline methods record zero. The static threshold performs better for point and collective anomalies, with F1-scores of 1.0000 and 0.6909, respectively. All three methods record zero for sensor drift. These findings highlight the strengths and limitations of each approach across different anomaly types.

Table 4. F1-Score Comparison Across Anomaly Categories

Anomaly category	Static Threshold (M1)	Context-Agnostic IF (M2)	CASC-MO Tuned (M3+)
Point anomalies	1.0000	0.1212	0.0879
Contextual anomalies	0.0000	0.0000	0.2523
Collective anomalies	0.6909	0.6190	0.4698
Sensor drift	0.0000	0.0000	0.0000

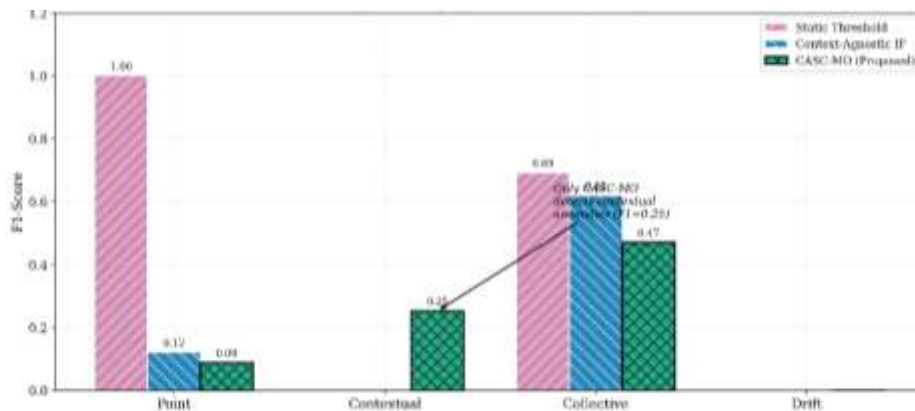


Figure 4. F1-Score Comparison Across Anomaly Categories

The figure 4 compares the F1-scores of the three detection methods across point, contextual, collective, and drift anomalies. The static-threshold method performs best for point anomalies and achieves the highest score for collective anomalies. The context-agnostic Isolation Forest also demonstrates comparatively strong performance for collective

anomalies. In contrast, CASC-MO is the only method that achieves a non-zero F1-score for contextual anomalies, suggesting that context-based model selection can help identify condition-dependent irregularities. However, all three approaches fail to detect drift anomalies effectively. These findings demonstrate that detection performance varies by anomaly category and that further improvements are needed for gradual behavioral changes.

4. Discussion

The results indicate that the proposed Context-Aware Sensor-Cloud Architecture with Integrated Machine Learning Analytics and Dynamic Model Orchestration (CASC-MO) offers advantages in identifying anomalies under different operating conditions. The proposed approach achieved an AUC-ROC of 0.8824 and a recall of 82.81%, outperforming the static-threshold method and context-agnostic Isolation Forest, which both achieved a recall of 43.75%. These findings suggest that combining contextual information with adaptive model selection can improve the identification of unusual sensor behaviour.

However, the improvement in recall came with a reduction in precision. CASC-MO recorded a precision of 0.4046 and a false-alarm rate of 0.0633, indicating that a proportion of normal observations were incorrectly classified as anomalies. The static-threshold method achieved perfect precision in this experiment but missed more anomalous observations. This trade-off highlights the importance of selecting detection thresholds according to application requirements. The broader outlier-detection literature similarly demonstrates that anomaly detection performance depends on the characteristics of the data and the evaluation criteria used (Wang *et al.*, 2019).

A notable result was the improvement in contextual anomaly detection. CASC-MO achieved an F1-score of 0.2523 for contextual anomalies, whereas both comparison methods recorded an F1-score of zero for this category. This suggests that incorporating operating context can help identify observations that are unusual relative to their circumstances, even when their individual values do not exceed fixed thresholds.

For example, a sensor reading that is reasonable during daytime activity may be unexpected during an inactive period. The context descriptor and nearest-neighbour selection mechanism are intended to account for such differences by choosing a model associated with a similar operating state. Nevertheless, the contextual F1-score remains modest, indicating that the approach does not yet provide consistently strong detection across all contextual cases. Research on time-series anomaly detection emphasizes the importance of evaluating methods against different anomaly types rather than relying on a single aggregate score (Schmidl *et al.*, 2022).

The category-level results reveal that CASC-MO does not perform equally well for every anomaly type. Its point-anomaly F1-score was 0.0879, compared with 1.0000 for the static-threshold baseline. For collective anomalies, CASC-MO achieved 0.4698, below the static-threshold score of 0.6909 and the context-agnostic Isolation Forest score of 0.6190. These findings show that the overall improvement in recall does not establish superiority across every category.

The results also indicate that drift anomalies were not detected successfully by any of the three methods. This limitation is important as gradual changes in sensor behaviour may not resemble isolated deviations or short groups of unusual observations. Time-series approaches based on temporal relationships and learned representations could provide useful directions for future investigation. Xu *et al.* (2021), for example, introduced the Anomaly Transformer, which uses association discrepancy to identify anomalous temporal patterns. Self-supervised learning is another relevant research direction for learning from time-series data where labelled anomalies are limited (Sánchez-Ferrera *et al.*, 2025). These approaches could be evaluated as potential extensions rather than assumed to improve CASC-MO without experimental validation.

CASC-MO recorded an average detection latency of 17.23 ms per observation. This result suggests that contextual model selection and inference may be practical for some sensor-monitoring applications, although deployment suitability depends on hardware, workload, data volume, and response-time requirements. The comparison methods had lower reported latency, so the adaptive approach introduces additional computational cost.

This consideration is particularly relevant to sensor networks where communication and processing resources may be constrained. Raza *et al.* (2017) discuss the characteristics and trade-offs of low-power wide-area networks, while Xavier *et al.* (2022) investigate a framework for in-network machine-learning-based traffic classification. These studies provide relevant background for considering where analytics should be executed and how resource constraints affect system design. Similarly, machine learning has been investigated for network assurance in intent-based data centres (Zheng and Leivadeas, 2021). Such work supports the broader relevance of intelligent monitoring, but it does not independently validate the scalability of CASC-MO.

Several limitations must be considered when interpreting these findings. First, the evaluation used a simulated smart-building dataset rather than measurements collected from a real deployment. Second, the reported results represent a single seeded experiment, so repeated trials and statistical significance testing are needed to establish the consistency of the observed differences. Third, the failure to detect drift indicates that the current design requires further development for evolving sensor behaviour.

Future work should evaluate the architecture using real-world datasets, multiple random seeds, and a wider range of operating conditions. It should also investigate drift-sensitive features, temporal modelling, and alternative anomaly-detection methods. Research on deep learning for prognostics and health management offers additional background

for studying equipment degradation and developing data-driven monitoring systems (Zhang *et al.*, 2019). Recurrent neural networks have also been investigated for intrusion detection, illustrating another application of learned temporal patterns to identifying abnormal behaviour (Yin *et al.*, 2017). Finally, emerging work on AI-native network operating systems highlights the importance of architecture, control loops, and governed autonomy when integrating intelligence into networked systems (Sahin *et al.*, 2026). These ideas may inform future extensions, but their effectiveness within CASC-MO must be established experimentally.

Overall, CASC-MO demonstrates promising gains in aggregate discrimination and recall, particularly for contextual anomalies. Its false-alarm rate, category-dependent performance, computational overhead, and inability to detect drift remain important challenges. The findings therefore support further investigation rather than a claim of universal superiority over existing detection methods.

5. Conclusion

This study presented CASC-MO, a Context-Aware Sensor-Cloud Architecture that integrates machine learning analytics and dynamic model orchestration for sensor anomaly detection. The proposed architecture combines context characterisation, context-specific model selection, and anomaly detection to identify unusual sensor behaviour under varying operating conditions. Its effectiveness was examined by comparing it with conventional detection approaches using a simulated smart-building dataset. The findings indicate that incorporating contextual information can improve the identification of anomalies that may be overlooked by methods relying on fixed thresholds or context-independent models. By adapting model selection to the prevailing operating context, CASC-MO provides a flexible approach to sensor monitoring. The architecture also offers a foundation for integrating intelligent analytics into sensor-cloud environments, where different operating conditions may require different detection strategies. Despite these benefits, several challenges remain. The approach did not successfully identify drift anomalies, and its detection performance varied across anomaly categories. The evaluation also relied on simulated data and a single experimental run, limiting the extent to which the findings can be generalised to real-world environments. Future studies should focus on validating the architecture using real-world sensor datasets, improving drift detection, reducing false alarms, and evaluating performance across repeated experiments and diverse operating conditions. Further investigation into computational efficiency and adaptive model selection could also support practical deployment. Overall, CASC-MO provides a promising direction for context-aware anomaly detection, with further validation required to establish its robustness and suitability for real-world sensor-cloud applications.

References

1. Alsharif, M. H., Kelechi, A. H., Yahya, K., & Chaudhry, S. A. (2020). Machine learning algorithms for smart data analysis in internet of things environment: taxonomies and research trends. *Symmetry*, 12(1), 88.
2. Arief-Ang, I. B., Hamilton, M., & Salim, F. D. (2018). A scalable room occupancy prediction with transferable time series decomposition of CO2 sensor data. *ACM Transactions on Sensor Networks (TOSN)*, 14(3-4), 1-28.
3. Benzaid, C., & Taleb, T. (2020). AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions. *Ieee Network*, 34(2), 186-194.
4. Bhatia, A., Khomh, F., Adams, B., & Hassan, A. (2026). An empirical study of self-admitted technical debt in machine learning software. *ACM Transactions on Software Engineering and Methodology*, 35(8), 1-41.
5. Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1), 1-99.
6. Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache Flink™: Stream and Batch Processing in a Single Engine. *IEEE Data Eng. Bull.*, 38(4), 28-38.
7. Cicirelli, F., Guerrieri, A., Mercuri, A., Spezzano, G., & Vinci, A. (2019). Itema: A methodological approach for cognitive edge computing iot ecosystems. *Future Generation Computer Systems*, 92, 189-197.
8. Cook, A. A., Misırlı, G., & Fan, Z. (2019). Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal*, 7(7), 6481-6494.
9. de Sousa, N. F. S., Perez, D. L., Rothenberg, C. E., & Gomes, P. H. (2021). End-to-end service monitoring for zero-touch networks. *Journal of ICT Standardization*, 9(2), 91-112.

10. Dromard, J., Roudière, G., & Owezarski, P. (2016). Online and scalable unsupervised network anomaly detection method. *IEEE Transactions on Network and Service Management*, 14(1), 34-47.
11. Fortino, G., Savaglio, C., Spezzano, G., & Zhou, M. (2020). Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(1), 223-236.
12. Gugulothu, N., Malhotra, P., Vig, L., & Shroff, G. (2018, July). Sparse neural networks for anomaly detection in high-dimensional time series. In *AI4IOT workshop in conjunction with ICML, IJCAI and ECAI* (pp. 1551-3203).
13. Guha, S., Mishra, N., Roy, G., & Schrijvers, O. (2016, June). Robust random cut forest-based anomaly detection on streams. In *International conference on machine learning* (pp. 2712-2721). PMLR.
14. Hariri, S., Kind, M. C., & Brunner, R. J. (2019). Extended isolation forest. *IEEE transactions on knowledge and data engineering*, 33(4), 1479-1489.
15. Jayalakshmi, M., Nyamasvisva, T. D. T. E., Buhari, A., Sobia, M. C., & Maharajan, K. (2026). A Context-Aware Sensor-Cloud Architecture with Integrated Machine Learning Analytics and Dynamic Model-Orchestration Framework. *International Journal of Artificial Intelligence and Machine Learning*, 6(10s), 118-137.
16. Kim, M. S., Yun, J. P., Lee, S., & Park, P. (2019, March). Unsupervised anomaly detection of lm guide using variational autoencoder. In *2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE)* (pp. 1-5). IEEE.
17. Liu, J., Shen, H., Narman, H. S., Chung, W., & Lin, Z. (2018). A survey of mobile crowdsensing techniques: A critical component for the internet of things. *ACM Transactions on Cyber-Physical Systems*, 2(3), 1-26.
18. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3), 12-22.
19. Murugesan, S. B. P., & Ganesan, D. (2026, January). AI-Native Observability and Self-Evolving Diagnostics: Generative Troubleshooting, Semantic Correlation, and Embedded Guidance for Large-Scale Cloud Systems. In *2026 6th International Conference on Image Processing and Capsule Networks (ICIPCN)* (pp. 1167-1172). IEEE.
20. Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. *IEEE access*, 9, 78658-78700.
21. Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low power wide area networks: An overview. *ieee communications surveys & tutorials*, 19(2), 855-873.
22. Sahin, O., Sathya, V., & Zhang, L. (2026). A Survey and Tutorial on AI-Native Network Operating Systems for 6G Communication Networks: Architectures, Control Loops, Memory, and Governed Autonomy. *Control Loops, Memory, and Governed Autonomy (July 23, 2026)*.
23. Sánchez-Ferrera, A., Calvo, B., & Lozano, J. A. (2025). A review on self-supervised learning in time series anomaly detection: Recent advances and open challenges. *ACM Computing Surveys*, 58(5), 1-35.
24. Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation.
25. Wang, H., Bah, M. J., & Hammad, M. (2019). Progress in outlier detection techniques: A survey. *Ieee Access*, 7, 107964-108000.
26. Xavier, B. M., Guimarães, R. S., Comarela, G., & Martinello, M. (2022). Map4: A pragmatic framework for in-network machine learning traffic classification. *IEEE Transactions on Network and Service Management*, 19(4), 4176-4188.
27. Xu, J., Wu, H., Wang, J., & Long, M. (2021). Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*.
28. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE access*, 5, 21954-21961.

29. Zhang, L., Lin, J., Liu, B., Zhang, Z., Yan, X., & Wei, M. (2019). A review on deep learning applications in prognostics and health management. *Ieee Access*, 7, 162415-162438.
30. Zheng, X., & Leivadeas, A. (2021, October). Network assurance in intent-based networking data centers with machine learning techniques. In *2021 17th International conference on network and service management (CNSM)* (pp. 14-20). IEEE.