Proceedings of the Pakistan Academy of Sciences: A Physical and Computational Sciences 63(3): ------- (2026)
Copyright © Pakistan Academy of Sciences
ISSN (Print): 2518-4245; ISSN (Online): 2518-4253
http://doi.org/10.53560/PPASA(63-3)720

Pakistan Academy of Sciences

Research Article

H-RADIS: An Adaptive Edge-Centric Lightweight Framework for Real-Time Multi-Stream Anomaly Detection in Smart Building IoT Systems

Mubashir Abbas[1]*, Humaira Ijaz[2], Muhammad Fahad[1], Bushra Jamil[2], Bilal Ashraf[2], and Kashif Munir[3]

[1]Department of Computer Science, University of Sargodha, Sargodha, Pakistan
[2]Department of Information Technology, University of Sargodha, Sargodha, Pakistan
[3]Department of Computer Science, FAST-NUCES, Islamabad, Pakistan

ACCEPTED MANUSCRIPT

Abstract: The rapid adoption of smart, real-time, QoS-aware IoT applications generates a massive amount of multi-stream data collected from heterogeneous interconnected devices. Extracting valuable and immediate insights from this data requires real-time processing and analytics. Real-time anomaly detection enhances these insights by identifying deviations in data as they occur, ensuring operational reliability, energy efficiency, and resilience against faults. Detecting anomalies in such streams involves analyzing individual streams and identifying complex interrelationships among multiple streams, where irregular patterns may emerge collectively or over time. Therefore, this paper proposes H-RADIS, an adaptive Half-Space Trees (HST) based framework specifically designed for real-time, multi-stream anomaly detection on edge devices. It effectively detects point and combined anomalies and employs Complex Event Processing (CEP) to detect collective anomalies. The framework integrates a sliding window-based dynamic threshold with adaptive parameter tuning, enabling the H-RADIS to cope with varying situations in real time more effectively. The performance of H-RADIS is evaluated against RIVER-based online machine learning algorithms, Local Outlier Factor (LOF), and OneClassSVM (OCSVM) across two different datasets by augmenting them with synthetic anomalies. The experimental results indicate that H-RADIS consistently improves anomaly detection performance over the collective LOF–OCSVM baseline, with relative gains in F1-score (26.08%), Recall (23.88%), Precision (22.22%), and ROC-AUC (17.69%). In terms of computational efficiency, H-RADIS achieves a relative 48.65% increase in throughput, together with relative reductions of 34.71% in processing time, 34.69% in energy consumption, and 0.21% in memory usage, while maintaining comparable CPU utilization. Overall, the comparative results indicate that, under the evaluated experimental conditions, H-RADIS is a resource-efficient approach for real-time anomaly detection in smart-building IoT environments.

Keywords: IoT, Streaming Data, Edge Computing, HST, Incremental Learning, Real-Time Anomaly Detection.

1. INTRODUCTION

The Internet of Things (IoT) refers to the interconnection of physical devices and sensors that share data to create smarter systems and optimized processes across different domains such as agriculture, healthcare, energy management, and smart cities. The main goal of IoT is to establish an interconnected communication infrastructure among devices, thereby optimizing business processes and supporting the development of smart systems [1]. The proliferation of smart, real-time, and QoS-sensitive IoT applications generates massive volumes of data [2], primarily comprising continuous, multivariate time-series streams collected from heterogeneous sensors and consisting of sequential observations indexed by timestamps and sampled at either regular or irregular intervals [3]. This time-series data requires immediate processing and real-time

Received: June 2026; Revised: August 2026; Accepted: September 2026
* Corresponding Author: Mubashir Abbas <mubashir.abas@gmail.com>

analytics to extract valuable knowledge and meaningful insights essential for intelligent decision-making in smart environments such as autonomous driving, healthcare monitoring, smart buildings, and emergency services [4]. Traditional cloud computing is not suitable for real-time analytics due to increased latency, which hinders prompt decision-making, crucial for time-sensitive tasks. This necessitates real-time data processing and analysis to minimize latency and maximize throughput.

To process and analyze real-time data closer to the data source in IoT environments, edge and fog computing are the most suitable choices. Benefits such as fast real-time processing, efficient bandwidth utilization, and improved security are enabled by edge computing, which processes data near the data source [5]. Edge devices are particularly well suited to real-time analytics for detecting equipment malfunctions, abnormal environmental conditions, unusual sensor patterns, and potential security breaches because local data processing reduces latency and enables prompt responses [6]. Such unusual patterns that do not match the expected norms are called anomalies, indicating an issue or rare event within the multi-stream data. These anomalies need rapid identification and further analysis to avoid critical system failures, optimize operational performance, and mitigate risks, thus making anomaly detection an indispensable aspect of comprehensive real-time analytics solutions [7]. The different types of anomalies are discussed below:

- **Point Anomalies:** A single data point substantially differs from the rest of the data. For example, a single patient's blood pressure that is notably higher than normal indicates a health issue.
- **Contextual Anomalies:** A data point deemed abnormal in one context can be normal in another. For example, recognizing irregular heart rates in the context of a patient's activity.
- **Collective Anomalies:** A series of data points from a single sensor over time or space which, when viewed together, are anomalous, even if individual points are not. For example, consecutive humidity values indicate a collective anomaly even if an individual reading is normal.
- **Combined Anomalies:** When fusion of data points from multiple sensors shows anomalies together, indicating an interrelation between different sensors that can cause system failures. For example, an unexpected escalation in both temperature and humidity indicates failure of an air conditioner.

Detecting different types of anomalies in multi-stream data at the edge level is challenging due to resource constraints in terms of processing power, memory, and storage [8]. Handling the diverse nature of data streams from various sensors also adds another layer of complexity as they vary in format, frequency, and quality. In such scenarios, sensor fusion plays a crucial role in integrating multi-source data into a unified format for applying different types of analysis to it. It enables a more efficient and reliable anomaly detection approach based on holistic insights of the system and pattern recognition that can be ignored while analyzing isolated streams [9]. These challenges necessitate the development of advanced real-time frameworks for efficient and scalable anomaly detection on edge computing, ensuring continuous monitoring of dynamic IoT multi-stream data.

To achieve the objective of real-time processing with high responsiveness and efficient resource utilization, many machine learning and deep learning techniques have emerged. Verdone *et al*. [10] proposed an anomaly detection approach for multivariate time series data to capture normal behavior and identify anomalies through self-supervised forecasting tasks. This approach has shown promising results even without expert knowledge, accommodating a variety of datasets. In the same way, PatchAD, an effective anomaly-detection model, was presented by Zhong *et al*. [11] using the MLP-Mixer architecture and optimized to run on resource-constrained edge devices. This method introduced the potential of lightweight deep learning models in anomaly detection tasks in IoT and edge computing systems. Based on current methods, Ayad *et al.* [12] introduced a lightweight model that combines a one-class asymmetric stacking autoencoder for dimensionality reduction and a one-class DNN for anomaly detection and achieved a significant detection rate of up to 96.27%, with an accuracy of 99.99%, precision of 99.21%, and an F1 score of 97.69%. In parallel, on-device AI models, their applications, optimization techniques, and future trends to facilitate smart intelligence

ACCEPTED MANUSCRIPT

in resource-constrained devices are discussed by Wang *et al*. [13]. To enable real-time monitoring and proactive anomaly detection in IoT applications, a secure IoT anomaly-detection framework was proposed by Prakash *et al*. [14]. Using AWS IoT Core infrastructure, this framework employed Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Gaussian Naive Bayes (GNB) on synthetic datasets with induced anomalies and achieved 91–98% detection accuracy, with CART providing the best overall performance for identifying anomalous behavior. Similarly, Islam *et al*. [15] proposed a hybrid fog-edge architecture to monitor health in IoMT systems by using rule-based filtering, statistical thresholds, Decision Trees, and One-Class SVMs. Compared to cloud-centric models, the architecture achieved a 70% reduction in latency, 60% in bandwidth consumption, and a 30% improvement in energy efficiency. Similarly, to address the energy efficiency problem, Ni *et al*. [16] proposed an optimization model to balance local processing and offloading in edge computing, reducing energy consumption by 23.8% while maintaining anomaly-detection accuracy above 92.5% across diverse anomaly types. In situ observational data, Chen *et al*. [17] evaluated unsupervised methods, including machine learning methods (LOF, IForest, LODA, OCSVM) and deep learning models (MSCRED, OmniAnomaly, Anomaly Transformer, Dual-TF) for anomaly detection. The authors introduced an enhanced MSCRED* model using a cosine-similarity matrix with POT-based threshold and achieved robust performance and improved interpretability of detected anomalies. For IoT traffic, Watanabe *et al*. [18] proposed a self-adaptive anomaly detection system by using a honeypot server to update the machine learning models in response to emerging cyber threats continuously. Results prove that dynamically updated models detect anomalous traffic more accurately than pre-trained models.

Ultimately, it is observed that despite recent advancements, real-time anomaly detection for heterogeneous IoT streams on resource-constrained edge devices remains challenging. Computationally intensive machine and deep-learning approaches may be unsuitable for continuous edge processing, whereas lightweight streaming methods often emphasize isolated point anomalies and provide limited support for interpreting the progression of abnormal behavior across time and across multiple sensor streams. In particular, the combined requirements of adapting to evolving data distributions, distinguishing isolated deviations from sustained malfunctions, identifying collective abnormalities across heterogeneous streams, and integrating these anomaly indications into an overall assessment of system behavior remain insufficiently addressed within a single lightweight streaming framework. Therefore, there is a need for an adaptive and resource-efficient multi-level anomaly-detection solution capable of progressing from point-level detection to collective, and combined system-level assessment in heterogeneous streaming IoT environments.

Smart-building HVAC systems generate continuous multi-stream data with changing operational patterns, making fixed thresholds and point-wise detection susceptible to false alarms, missed detections, and overlooked temporal relationships. Online algorithms available in the River framework [19], including Half-Space Trees (HST) [20], Local Outlier Factor (LOF) [21], and One-Class Support Vector Machine (OCSVM) [22], provide incremental processing without repeated batch retraining that provide basis for real-time processing on resource constraint devices. Among these, HST is particularly suitable for high-velocity streaming data because of its incremental operation and relatively low computational overhead. However, conventional HST primarily produces point-level anomaly scores and does not inherently provide adaptive multi-level detection over local sensor deviations, collective cross-stream behavior, and overall system abnormality. This limits its ability to support multi-type anomaly detection in dynamic smart-building environments.

Accordingly, this study proposes H-RADIS, an adaptive HST-based Real-time Anomaly Detection framework for IoT Systems, deployed at the edge layer of a smart-building network. The methodological novelty of H-RADIS lies in combining HST, dynamic threshold, Complex Event Processing (CEP), and anomaly fusion, and also organizing these mechanisms into unified multi-level streaming decision architecture. H-RADIS first performs point-level anomaly detection on individual sensor streams using HST-supported online scoring and sliding-window-based adaptive thresholds. It then evaluates temporal persistence

ACCEPTED MANUSCRIPT

to distinguish isolated deviations from sustained abnormal behavior by using CEP-based mechanism. These anomaly indications are subsequently integrated to provide a combined system-level assessment of abnormal behavior. Through sensor-specific and joint adaptive decision mechanisms, the framework is designed to accommodate evolving stream characteristics while retaining the lightweight incremental processing required for edge-oriented deployment. The major contributions of this research are as follows:

1. We developed an edge-centric framework that progresses from point-level anomaly detection to sustained abnormality identification, collective multi-sensor anomaly interpretation, and combined system-level assessment, thereby providing a broader view of system behavior than isolated point-wise detection alone.
2. We designed H-RADIS by extending HST with sliding-window-based dynamic threshold, and parameter adaptation to better accommodate changing operational patterns in heterogeneous streaming environments. The framework further incorporates temporal persistence and CEP-based collective-event logic to distinguish isolated point anomalies from persistent malfunctions and broader multi-sensor abnormal events.
3. H-RADIS is implemented using incremental online processing to support responsive anomaly detection on resource constraint devices. Its performance is evaluated across two datasets and compared with River-based LOF and OCSVM baselines in terms of anomaly-detection effectiveness, cross-dataset consistency, statistical significance, responsiveness, and computational resource utilization. The reported results indicate favorable performance and computational behavior for H-RADIS under the evaluated experimental conditions.

2. MATERIALS AND METHODS

To evaluate the practical applicability of the proposed real-time anomaly detection framework under realistic operating conditions, this study considers an integrated HVAC system in a smart building environment as a representative case study.

2.1. Case Study: Integrated HVAC Systems in Smart Buildings

Smart buildings are an integral part of architectural design and management in smart ecosystems. In a smart building ecosystem, various devices, sensors, and automated systems are integrated to provide efficient, responsive, and intelligent services. This integration of multiple subsystems like HVAC, security, and energy management systems into a single system provides the basis for real-time anomaly detection and analysis of continuously generated data for automatic control of operations based on varying conditions and occupant needs. The HVAC main components include indoor climate, air quality index, light, temperature, humidity level, and energy optimization. For HVAC systems, real-time anomaly detection is very important, which can identify minor deviations and major faults, to enable optimal performance of smart building management and control systems. It also helps to minimize failure rate, maintaining a healthier and more productive indoor environment to ensure an adaptive occupant's comfort level. In the long term, real-time anomaly detection supports the sustainability and cost-effectiveness of smart building systems, which in turn supports economic growth, environmental efficiency, and overall productivity. This situation underlines the necessity of a more holistic anomaly-detecting strategy that is capable of being more responsive to detect real-time point, collective, and multi-sensor anomalies and a more adaptive system control to enhance the reliability and energy efficiency of HVAC system operation.

2.2. H-RADIS (HST-Based Real-Time Anomaly Detection Framework for IoT Systems)

To develop an edge-centric, advanced anomaly detection solution for multi-stream IoT data that optimizes responsiveness while minimizing computational overhead in resource-constrained environments, an H-RADIS framework, shown in Figure 1, is proposed. H-RADIS integrates adaptive HST with sliding-window-based dynamic thresholds and adaptive parameter tuning to effectively handle evolving data patterns in streaming scenarios for real-time anomaly detection and employs Complex Event Processing (CEP) for collective anomaly detection that facilitates detection of isolated faults

ACCEPTED MANUSCRIPT

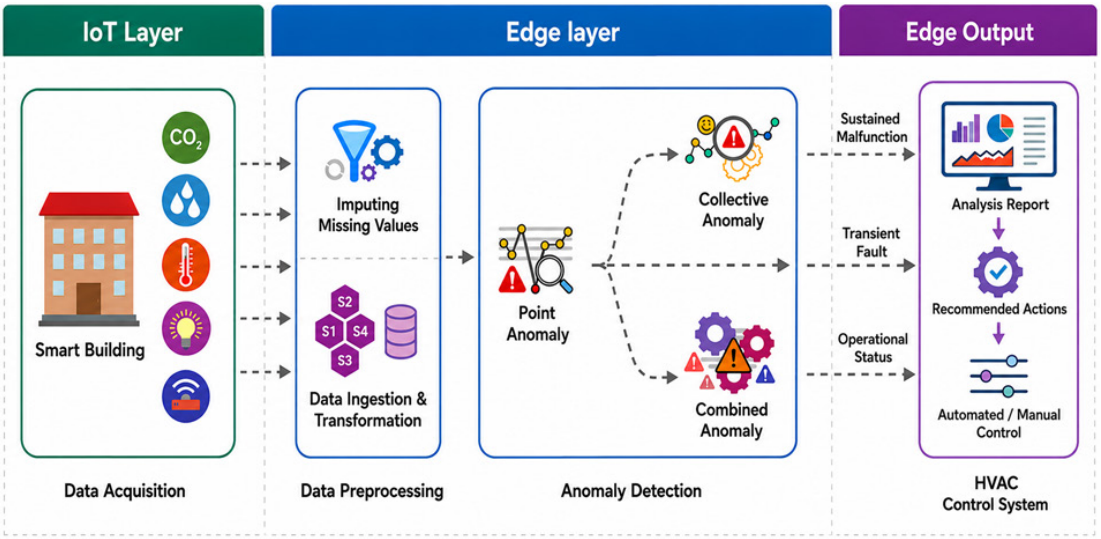

Fig. 1. An overview of the H-RADIS framework.

from sustained malfunctions. Moreover, combined anomalies are detected by fusing different sensor streams to provide a broad system-level assessment of abnormal behavior. In functionality, H-RADIS is structured around two main processing layers: the IoT layer, responsible for real-time data acquisition from sensors, and the edge layer, which handles data preprocessing and anomaly detection. The results generated at the edge are then forwarded to the HVAC control system to enable informed decisions based on the type and severity of detected anomalies. Detailed functionality of the proposed framework is discussed as follows.

2.2.1. Data acquisition

Real-time HVAC data consisting of temperature, humidity, light, gas, and accelerometer streams is continuously collected through a testbed deployed in an office environment. These sensors continuously generate these streams at regular intervals, exhibiting the behavior of the indoor environment.

2.2.2. Preprocessing

As raw sensor data can often be noisy or incomplete, initial preprocessing is required to impute missing values, align timestamps, and bring the data into a consistent and structured format to get reliable analysis results. Before applying the anomaly detection model, the following two key preprocessing steps are performed:

2.2.2.1. Data ingestion and transformation:
The HVAC real-time data is collected through sensors, and their values are stored in a CSV file. Sensor data for further processing is fetched from a CSV file and pivoted into the new format as per the model requirement.

2.2.2.2. Imputing missing values:
One main factor that can compromise the model's performance is missing data. To maintain data continuity and ensure reliable anomaly detection, a "mean" strategy is used to fill the missing values.

2.2.3. Anomaly detection with H-RADIS

To detect point, collective, and combined anomalies across multiple IoT data streams more systematically, H-RADIS employs HST as a backbone model with certain variations like a sliding window, dynamic threshold with adaptive parameter tuning, dynamic scaling, and CEP (Complex Event Processing), which are discussed below:

2.2.3.1. Half-Space Trees (HST):
The proposed H-RADIS framework includes Half-Space Trees (HST) as the core anomaly detection model. The HST, introduced by Tan et al. [20], is an isolation-based ensemble technique for anomaly detection in time-varying data streams. It divides the feature space into regions by randomly sampling half-space boundaries and then gives higher anomaly scores to observations in sparsely populated regions, which are not in line with the data distribution. The tree structures built by HST are built autonomously from the training

ACCEPTED MANUSCRIPT

data, making it possible to process streaming observations with constant average processing time and memory consumption [20]. Such features make it applicable in resource-limited edge environments for real-time anomaly detection. H-RADIS uses a machine-learning library called RIVER, which enables incremental processing of dynamic data streams, one observation at a time. The HST implementation under the RIVER and PySAD frameworks was subsequently evaluated by Ray and Susan [23] through a comparative analysis of different online anomaly-detection methods. Thus, HST offers H-RADIS with an efficient online anomaly detection foundation, which is further strengthened by using adaptive threshold and complex-event processing (CEP).

2.2.3.2. HST parameters:

The performance of HST against dynamic requirements in terms of accuracy, responsiveness, and resource utilization significantly depends on its default parameters and attributes, where the role of n_{trees}, *height*, and *limits* is important with respect to accuracy and resource utilization. A higher value of accuracy can be obtained through increasing the value of n_{trees} that may cause higher computational cost. Then, to control the depth of the tree for capturing more detailed information, *height* parameter is managed, while the granularity of the partition process depends on the *limit* parameter. It defines a threshold value for the number of data points that are to be stored on each node before partitioning. Moreover, the responsiveness of HST is managed through *window_size* parameter, where a smaller *window_size* is used for higher responsiveness and vice versa. Another important attribute that belongs to memory usage and processing capability of HST is *size_limit*. It helps to keep the model lightweight by controlling the number of elements stored by each tree in HST. In the end, a specific value of *seed* parameter is used to control randomization and produce the same results across all executions. In H-RADIS, HST is used with following parameters configuration given in Table 1.

2.2.3.3. Dynamic threshold with adaptive parameter tuning:

The H-RADIS uses a dynamic threshold with adaptive parameter tuning to make HST adaptive, responsive, and accurate with minimum false-positive and false-negative rates. Basically, dynamic threshold is a decision boundary calculated from the recent anomaly score distribution to adapt the detection process to streaming data. Adaptive threshold parameter tuning is the process of automatically adjusting the parameters of the dynamic threshold according to the statistics of anomaly scores. In H-RADIS, the mean μ_s and standard deviation σ_s of the recent anomaly score window are used to calculate the dynamic threshold value $\theta adaptive = (\mu_s + k\sigma_s) / (1 + c\sigma_s)$, where the sensitivity multiplier k is set to 2.0, 2.5, or 3.0 depending on the score variability and the stabilization parameter c is set to 0.3, 0.5, or 0.7 depending on the mean score level. Similarly, the quantile parameter q is set to 0.85, 0.80, or 0.75 to update the quantile filter according to the score variability. In H-RADIS, HST continuously calculates the mean and standard deviation of

ACCEPTED MANUSCRIPT

Table 1. Configuration of HST parameters under H-RADIS framework.

Component	Parameter	Selected Value	Function
HST	n_{trees}	5	Specifies the number of trees in the ensemble
	Height	5	Determines the maximum depth and partitioning resolution of each tree
	window_size	40	Controls how frequently HST updates its mass estimates
	Seed	42	Ensures reproducibility of random tree construction
Preprocessing	Scaler	Incremental MinMaxScaler	Normalizes streaming features online
Model Initialization	Warm-up observations	50	Initializes the scaler and HST before evaluation
Feature Smoothing	EWMA coefficient (α)	0.2	Controls the contribution of the current observation to the smoothed value

recent anomaly scores and updates its threshold values accordingly. The high value of multiplier k is increased for a higher level of sensitivity, while the values of c and q are adjusted according to the k factor. The complete procedure to calculate the dynamic threshold and apply adaptive parameter tuning are explained in algorithm 1.

2.2.4. Point anomaly detection with H-RADIS using dynamic threshold

H-RADIS leverages adaptive HST to detect point anomalies by examining each HVAC stream independently and identifying outliers by recursively partitioning the feature space. Afterwards, H-RADIS calculates the anomaly score for each data point and compares it with the dynamic threshold. If the anomaly score is greater than the threshold value, it is declared an anomaly and a normal data point otherwise. Applying this process to all streams enables the detection of point anomalies (spikes/drops) across the HVAC system. As a whole, finding these real-time point anomalies enables the HVAC control system to proactively identify certain abnormalities that can lead to sensor faults or immediate disruptions in the system. The way point anomalies are detected by H-RADIS is explained in Algorithm 2.

Algorithm 1. Dynamic threshold with adaptive parameter tuning.

1. Input: Recent anomaly scores S = {s1, s2, s3, ... sn}
2. Output: Updated Parametrs k, c, q, dynamic_threshold $\theta_{adaptive}$
3. Compute μ_s = mean(S) and σ_s = std (S)
4. Adjust k: Sensitivity Multiplier based on σ_s
if σ_s > 0.05 Then k = 3.0
else if σ_s > 0.02 Then k = 2.5
else k = 2.0
5. Adjust c: Smoothing Factor based on μ_s
if μ_s < 0.5 Then c = 0.3
else if μ_s < 0.7 Then c = 0.5
else c = 0.7
6. Adjust q: Quantile Threshold based on σ_s
if σ_s > 0.1 Then q = 0.75
else if σ_s >0.05 Then q = 0.80
else q = 0.85
7. Update Quantile Filter
quantilefilter ← QuantileFilter(Model M, q)
8. Computer dynamic_threshold $\theta_{adaptive}$:

$$\theta_{adaptive} = \frac{\mu_s + k.\,\sigma_s}{1 + c.\,\sigma_s}$$

9. Returned Parametrs k, c, q, dynamic threshold $\theta_{adaptive}$

Algorithm 2. Point anomaly detection with H-RADIS.

Input: Sensor data stream X_i , window size w , multiplier k
Output: dynamic threshold $\theta_{adaptive}$, Anomaly status$_i$ (t)
Initialize Empty queue Q of maximum length w
Initialize Model M for anomaly detection
for each data point X_i (t) in X_i (t) do
    Compute anomaly score S_i (t) using Model M
    Enqueue S_i (t) into Q , dequeue if Q exceeds w
    Update μ_{recent} and σ_{recent} from Q
    Calculate $\theta_{adaptive}(t) = \mu_{recent} + k \cdot \sigma_{recent}$
    if $S_i(t) > \theta_{adaptive}(t)$ then
        Anomaly$_i$ (t) ← 1
    else
        Anomaly$_i$ (t) ← 0
    end if
end for

2.2.5. Collective anomaly detection with H-RADIS using complex event processing (CEP)

H-RADIS extends its anomaly detection capabilities by integrating complex event processing (CEP) to detect collective anomalies. CEP identifies the interdependencies among various events and distills the most significant insights. By detecting these anomalies, sustained deviations or malfunctions can be identified in the HVAC system. These insights enable the framework to identify proactive maintenance tasks or systematic malfunctions. CEP works through a rule chain to detect collective anomalies as follows:

1. **Single anomaly rule (E1):** This rule triggers to identify a data point as anomaly when its anomaly score exceeds the adaptive threshold. This data point is stored in the CEP buffer along with its timestamp.
2. **Frequency rule (E2):** This rule checks for existence of at least n (e.g., n = 3) data points in CEP buffer. In case of less value of n, the event is marked as detection of isolated anomalous data points.
3. **Temporal continuity rule (E3):** This check ensures that time interval between each pair of consecutive anomalies must should be continuous. If this condition is violated, the sequence is considered broken and the buffer is reset.

On satisfying above CEP rules (E1, E2, E3), the buffered data points sequence is classified as a collective anomaly. Moreover, In H-RADIS, the CEP mechanism was implemented by setting a minimum threshold of three consecutive data points (n = 3) with temporal continuity to define a sequence as collective anomaly. The anomaly score threshold was initialized to 0.80 and updated using a sliding window of w = 60 most recent anomaly scores with k = 2.5, which controlled sensitivity to anomalies, and c = 0.5, which smoothed the threshold function. The complete procedure for identifying collective anomalies by H-RADIS using CEP is given in Algorithm 3.

2.2.6. Combined anomaly detection with H-RADIS using dynamic threshold and scaling

Combined anomaly detection can be expressed as one of the most innovative and significant anomaly detection tasks under the H-RADIS framework that simulates the concept of sensor fusion to integrate the anomaly score of all streams to obtain more intensive and productive insights. For detecting combined anomalies, H-RADIS calculates the anomaly score and assigns a weight to the anomaly score of each stream: temperature, humidity, light, gas, and accelerometer. Then, these weighted anomaly scores are aggregated to obtain

ACCEPTED MANUSCRIPT

Algorithm 3. Collective anomaly detection with H-RADIS using CEP.

Require: Sensor data stream X_i, window size w, threshold parameters k, c, min frequency n, max continuity gap Δt
Ensure: Detected Collective Anomaly Sequences Seq_i

Initialize: Recent score queue $QC \leftarrow \emptyset$, CEP anomaly buffer $C \leftarrow \emptyset$, last anomaly timestamp $t_{last} \leftarrow \emptyset$, Adaptive threshold $\theta_{adaptive} \leftarrow 0.8$
for each data point $X_i(t)$ in $X_i(t)$ do
  Calculate anomaly score $S_i(t)$and update Model M
  Enqueue $S_i(t)$ into Q , dequeue if $|Q| > w$
  Update $\theta_{adaptive}$ using μ_{recent} and σ_{recent} from Q
  Calculate $\theta_{adaptive}(t) = \mu_{recent} + k \cdot \sigma_{recent}$
  if $S_i(t) > \theta_{adaptive}(t)$ then
    (E1) Anomaly Detected
    Append $\{t, X_i(t)\}$ to CL
      if $t_{last} \neq \emptyset$ and $(t - t_{last}) > \Delta_t$ then
        Reset CL
      end if
    $t_{last} \leftarrow t$
  else
    Evaluate CEP Chain
    if $|Q| \geq n$ then
      (E2) Frequency condition satisfied
      (E3) Temporal Continuity satisfied
      Mark CL as Collective Anomaly Sequence Seq_i
    end if
    Reset CL
    $t_{last} \leftarrow \emptyset$
  end if
end for

a single value of anomaly score, which serves as the anomaly score of the overall system. This aggregated anomaly score is compared with the dynamic threshold to identify those data points as a combined anomaly. As this aggregated anomaly score contains the impact of all streams, it provides a clear reflection of the system's operational status. Ultimately, the detection of combined anomalies unveils another dimension of real-time monitoring that enables the identification of the actual situation of the system performance and then takes the correct measures accordingly. Equation (1) describes the calculation of the combined anomaly score A as follows:

$$A = \alpha \sum i \in \{T,H,L\}\, w_i f(i) + \beta \sum i \in \{G,A\} w_i f(i) + \gamma \sum i \in \{T,H,L,G,A\} w_i f(i) \quad (1)$$

Where $f(i)$ and w_i denote the anomaly score and weight, respectively, for each sensor stream (i). The streams are represented by temperature (T), humidity (H), light (L), gas (G), and accelerometer (A). Scenario weights α, β, and γ act as high-level fusion coefficients that regulate the relative influence of different sensor-group anomaly contributions in the final system-level anomaly score. Specifically, α controls the emphasis on environmental anomalies (temperature, humidity, and light), β prioritizes safety and disturbance-related anomalies (gas and accelerometer), and γ captures holistic system abnormality by incorporating all streams jointly. By tuning these parameters, typically under the constraint $\alpha + \beta + \gamma = 1$, the aggregation mechanism becomes scenario-aware and can adapt its sensitivity to operational priorities such as comfort-focused monitoring, safety-critical monitoring, or balanced system-health assessment. Algorithm 4 explains the complete procedure of detecting combined anomalies.

2.2.6.1. Dynamic Scaling:

In addition to a dynamic threshold with adaptive parameter tuning and CEP integration, another significant enhancement achieved by H-RADIS is the dynamic scaling. The key idea behind dynamic scaling is to normalize the combined anomaly score within a specific window of recent values. By maintaining a sliding window of recently combined anomaly scores, we dynamically adjust the minimum and maximum values for scaling, thereby ensuring that even small variations in anomaly scores are highlighted in the visualization. The mechanism of real-time monitoring and operational decision-making in smart building environments can be further improved with the detection and visualization of a combined anomaly score using dynamic scaling. The complete procedure for applying dynamic scaling on combined anomaly scores is given in Algorithm 5.

2.3. Experimental Setup

To evaluate H-RADIS under realistic operating conditions, a lightweight IoT-based smart building testbed was developed using an edge computing setup installed in an office environment. The sensing unit is built on an ESP32 microcontroller integrated with multiple sensors, including DHT22 for temperature and humidity, an MQ-series sensor for gas detection, an LDR for light intensity, and a PIR sensor for motion detection. This setup is shown

Algorithm 4. Combined anomaly detection with H-RADIS.

Input: Sensor data streams {X1, X2, ..., Xn}, window size w, multiplier k, weights {w1,w2,...,wn}
Output: dynamic_threshold $\theta_{adaptive}$, Anomaly status $_i$ (t),
Combined Anomaly Score $S_{combined}$ (t)
Initialize queues Q_i for each stream X_i of length w
Initialize Model M for anomaly detection
for each data point X_i (t) in X_i do
    Compute anomaly score S_i (t) using Model M
    Enqueue S_i (t) into Q_i, dequeue if Q_i exceeds w
    Update μ_{recent} and σ_{recent} from Q_i
    Calculate $\theta_{adaptive}(t) = \mu_{recent} + k \cdot \sigma_{recent}$
    Set Anomaly$_i$ (t) = 1 if $S_i(t) > \theta_{adaptive}(t)$, else 0
end for
Compute Combined Anomaly Score:
$$S_{combined}(t) = \sum_{i=1}^{n} x_i \cdot S_i(t)$$
Set Overall Status based on $S_{combined}$ (t)

ACCEPTED MANUSCRIPT

Algorithm 5. Dynamic scaling of combined anomaly score.

Input: Anomaly score stream A, window size w
Output: Scaled anomaly score a_{scaled}
Initial of an empty Q of length w.
Initialize $min_{window} \leftarrow \infty$
Initialize $max_{window} \leftarrow -\infty$
for each incoming anomaly score a_i in A **do**
    Q.enqueue (a_i)
    if Q.size() > w **then**
        Q.deque ()
    end if
    $min_{window} \leftarrow$ min (Q)
    $max_{window} \leftarrow$ max (Q)
        if $min_{window} = max_{window}$ **then**
            $a_{scaled} = 0.5$
        else
            $$a_{scaled} = \frac{a_i - min_{window}}{max_{window} - min_{window}}$$
        end if
end for

in Figure 2. The sensing node (microcontroller ESP-32) collects environmental data at regular intervals and transmits it to the Raspberry Pi (edge device) over a local wireless network using the MQTT protocol. Each data instance contains five features: temperature, humidity, gas value, light intensity, and motion status, and is stored in a CSV file with timestamps and sensor values, creating a stream of data under controlled conditions. The Raspberry Pi 3 Model B+, running Debian GNU/Linux 13.2 (Trixie) and equipped with a quad-core ARM Cortex-A53 processor based on the ARMv8-A (AArch64) architecture, operating at frequencies between 600 MHz and 1.4 GHz, and 1 GB of LPDDR2 RAM. A lightweight graphical environment (LXDE/Openbox) managed by LightDM was enabled to support live streaming and real-time anomaly detection visualization for HVAC streams. This architecture provides low-latency communication and does not require cloud infrastructure, so it is suitable for edge-based evaluation.

The anomaly detection models were implemented using the RIVER framework [19], which is specifically designed to process streaming data using incremental learning. Using RIVER, a single data point is processed or analyzed by the model at a time and updates its status continuously, enabling it to adapt to changing patterns of data without having to retrain on entire datasets. In this way, a real IoT smart environment scenario is created, whereby data is generated and processed in real-time, and decisions need to be made accordingly. The proposed and baseline models are tested on two different datasets to ensure both controlled validation and generalization. Detail information on both datasets is given in Table 2.

2.3.1. Primary dataset (dataset-1)

The first dataset, which is used as the primary dataset, is collected through a real-time testbed installed in an office environment, shown in Figure 2. The testbed was operated for one month (October to November 2025) and continuously recorded temperature, humidity, gas concentration, light intensity, and motion events. The observations were time stamped and sampled every 10 seconds, and stored in CSV format, creating a 12 MB multi-stream dataset. This controlled deployment enabled continuous operational data to assess the accuracy, responsiveness, and computational efficiency of the models in realistic edge-based monitoring scenarios, while simultaneously detecting anomalies.

2.3.2. Secondary dataset (dataset-2)

The secondary dataset is a publicly available smart building dataset [24] commonly used for tasks like anomaly and drift detection, etc. The smart building data (temperature, humidity, gas, light, and accelerometer, etc.) is collected from February 2022 to September 2022. Sensor values

ACCEPTED MANUSCRIPT

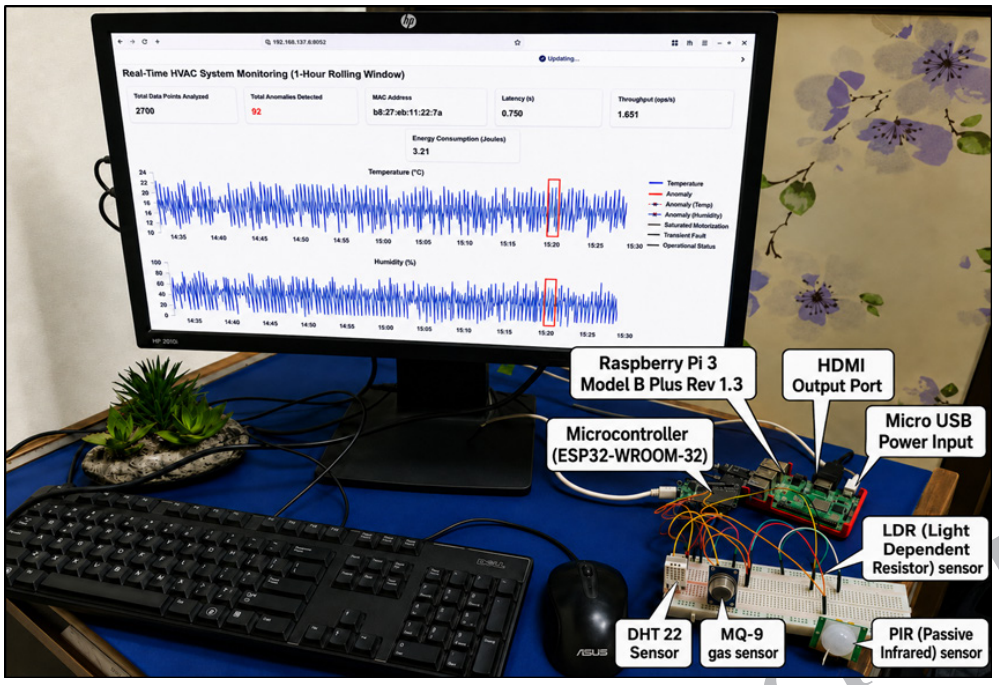

Fig. 2. Testbed setup for real-time anomaly detection in HVAC data streams.

ACCEPTED MANUSCRIPT

Table 2. Characteristics of the datasets used for H-RADIS evaluation.

Dataset	Data Source	Sensor Streams	Observation Period	Sampling Interval	Data Format	Storage Size
Multi-Sensor, Multi-device Smart Building Indoor Environmental Dataset	Historical Smart Building Dataset	Temperature, Humidity, Gas, Accelerometer, and Light	6 Months (Feb - Sep 2022)	10 Seconds	CSV	2.2 GiB
Real Testbed	Operational Smart Building Testbed	Temperature, Humidity, Gas, Light and, Motion Detection	1 Month (Oct - Nov 2025)	10 Seconds	CSV	12 MB

are measured every 10 seconds, and each record is time-stamped, containing device ID, sensor type, and measurement value, which is stored in a CSV file. This dataset is more varied and complicated, allowing us to check the generalization of models in real-time.

2.3.3. Real-Time execution on testbed data

H-RADIS was implemented following River's online learning paradigm, where incoming sensor observations are processed sequentially, single instance at a time, without batch retraining. The framework was executed directly on HVAC data collected from the real smart-building testbed as shown in Figure 2. This real-time execution enabled continuous multi-sensor streams to be analyzed under actual streaming conditions. During this execution, H-RADIS identified point, collective, and combined anomaly decisions in real time, thereby demonstrating its operational capability for continuous edge-oriented monitoring. However, the naturally occurring anomalies in the testbed were not independently labeled, and therefore these real-time detections could not be used as formal ground truth for quantitative performance evaluation. For this reason, controlled synthetic anomaly injection was subsequently used to create known anomaly labels for reproducible comparison of H-RADIS with the baseline methods.

2.3.4. Synthetic anomaly injection

To quantitatively evaluate anomaly-detection performance under controlled and reproducible conditions, the both datasets were augmented with a new "anomaly" column. Different percentages of synthetic anomaly injection (1%, 2%, 3%, and 5%) were experimented and ultimately, 5% and 3% were injected into dataset-1 and dataset-2, respectively, using a Python script that applies a random multiplier of ±30% to the original values, creating significant and detectable deviations. This helped in distinguishing anomalous behavior without completely distorting the nature of the data. This approach is similar to synthetic outlier generation techniques that use arithmetic mean, geometric mean, median, local scale, global scale, and magnitude warping, etc., that are implemented in an evaluation study by Kim and Joe [25] to improve the anomaly detection performance on time series datasets.

2.4. Baseline Models

The performance of H-RADIS is measured against two RIVER-based One-Class SVM (OCSVM) and Local Outlier Factor (LOF) models. Details of the selected baseline models are as follows:

2.4.1. Local outlier factor (LOF)

It is an online adaptation of the Local Outlier Factor (LOF) under the RIVER framework for streaming tasks [19]. It identifies outliers by comparing the local density of a data point to its neighboring points. LOF is effective for detecting local anomalies within unevenly distributed datasets but has higher computational complexity, which can be mitigated in large-scale systems through ensemble methods such as iForest [21].

2.4.2. One-Class SVM (OCSVM)

One-Class SVM (OCSVM) is an online machine learning algorithm that follows the RIVER framework pipeline for anomaly detection [19]. Basically, OCSVM identifies complex non-linear boundaries between normal and anomalous data, making it suitable for IoT environments with high device heterogeneity and varied traffic patterns. OCSVM constructs a decision boundary in a high-dimensional space, capturing normal behavior and flagging deviations as anomalies [22].

2.5. System Output

This section presents the visualization outputs on point, collective, and combined anomaly detection generated by H-RADIS, LOF, and OCSVM as follows:

2.5.1. Point anomaly

Point anomalies are characterized by sudden, isolated deviations from the normal data trend, often appearing as distinct spikes or drops in the time series data. H-RADIS marks point anomalies by isolating large deviations from the normal temperature range, such as a significant spike and drops around 00:30 time interval shown in Figure 3, indicating a transient fault in the temperature stream. On the other hand, LOF marks smaller

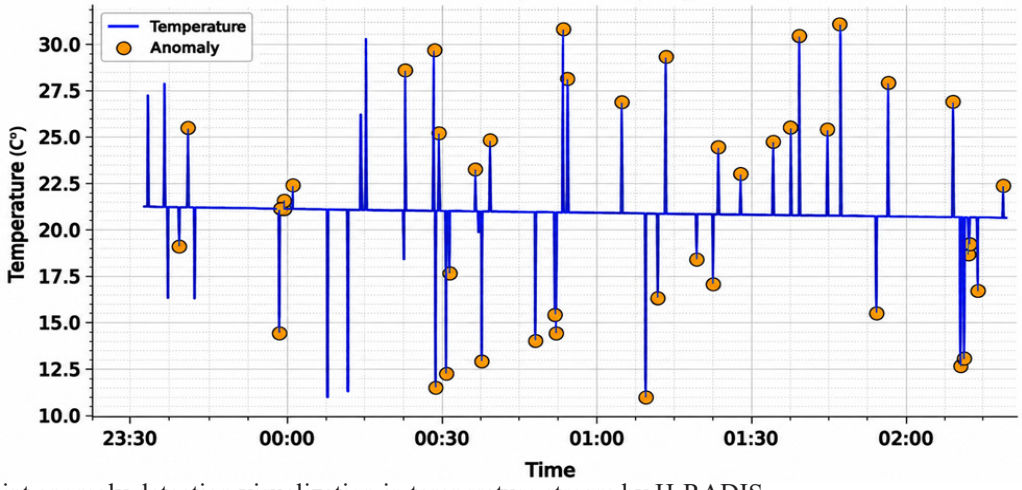

Fig. 3. Point anomaly detection visualization in temperature stream by H-RADIS.

ACCEPTED MANUSCRIPT

fluctuations as point anomalies based on changes in local density, like data points from 01:30 to 02:00 in Figure 4 are identified as point anomalies by the LOF. In contrast, OCSVM marks point anomalies by defining a boundary for normal data, declaring all points outside this boundary as anomalies, like the temperature spikes before and after the 00:30 time interval in Figure 5 is marked as point anomalies, as these data points are identified outside the normal range by the OCSVM. Overall, it is observed that H-RADIS detects and marks point anomalies more effectively than LOF and OCSVM.

2.5.2. Collective anomaly

Collective anomalies reflect consecutive deviations of data points from normal sensor behavior, forming a sustained abnormal pattern rather than isolated outliers. As shown in Figure 6, H-RADIS detects frequent collective anomaly patterns in gas data streams, marked with red vertical bars, particularly from 11:30 to 13:00, reflecting the occurrence of sustained faults during these time intervals that may lead to consistent abnormal behavior. In contrast, LOF primarily detects collective anomalies only in the earlier segment of the gas stream data, such as the data points around 23:40 are marked as collective anomalies, as shown in Figure 7. Collective anomaly pattern detection only in the early part of the time series data reflects LOF's limited ability to capture collective anomalies appropriately. While OCSVM captures fewer collective deviations, particularly in the region around 11:30 and 12:32, as illustrated in Figure 8. As a whole, H-RADIS has shown superior performance over LOF and OCSVM in detecting collective anomaly patterns in time series data, highlighting its ability to detect continuous anomalous patterns along with sudden spikes or drops only.

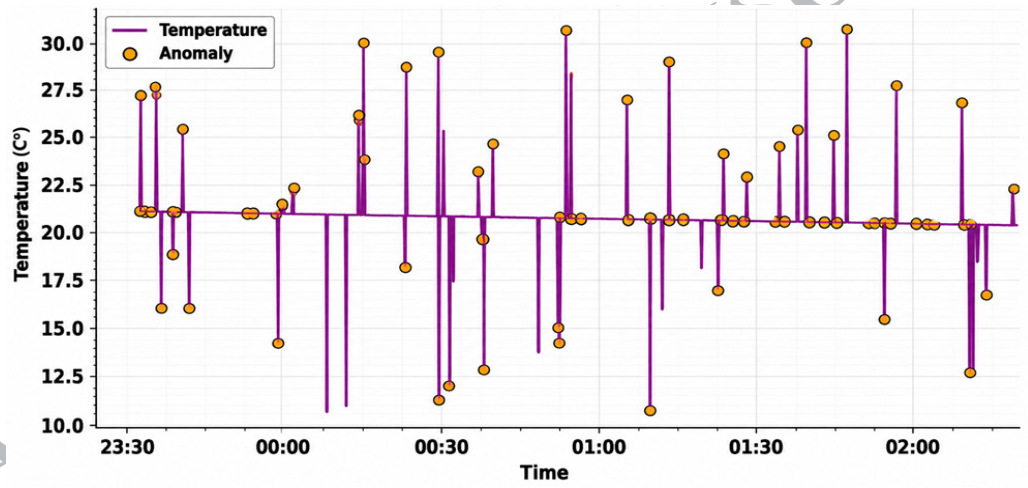

Fig. 4. Point anomaly detection visualization in temperature stream by LOF.

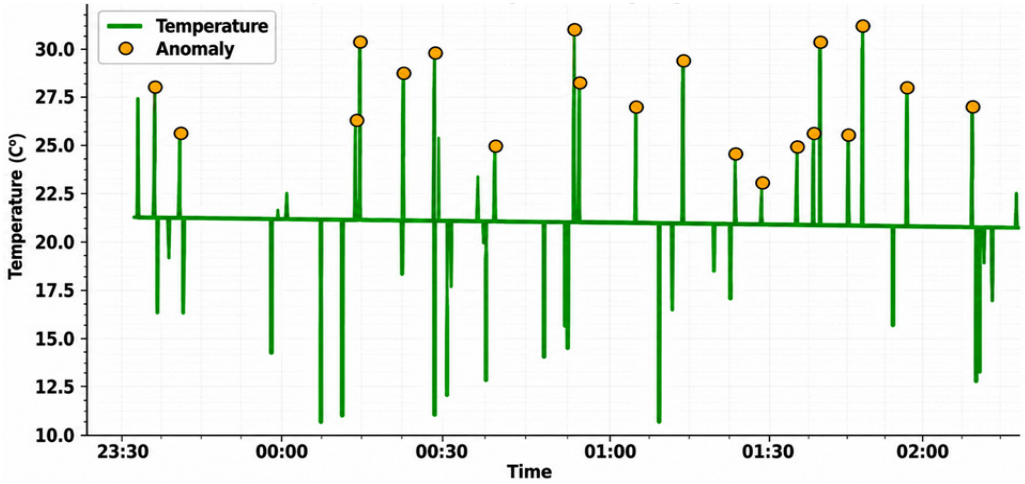

Fig. 5. Point anomaly detection visualization in temperature stream by OCSVM.

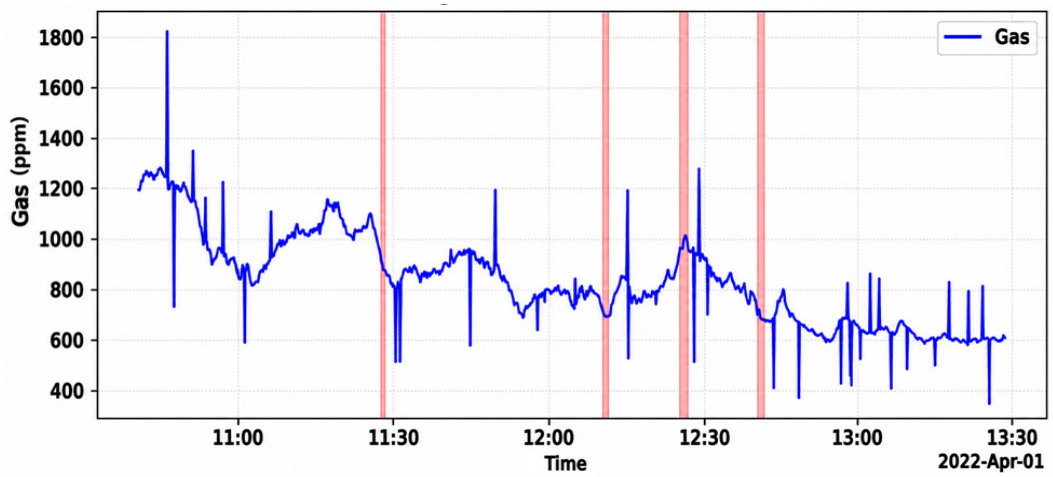

Fig. 6. Collective anomaly detection visualization in gas stream by H-RADIS.

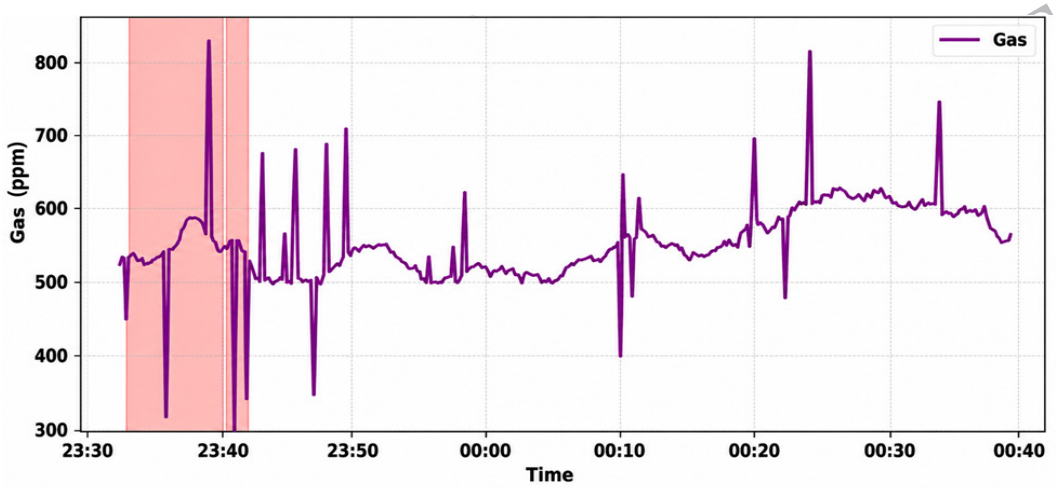

Fig. 7. Collective anomaly detection visualization in gas stream by LOF.

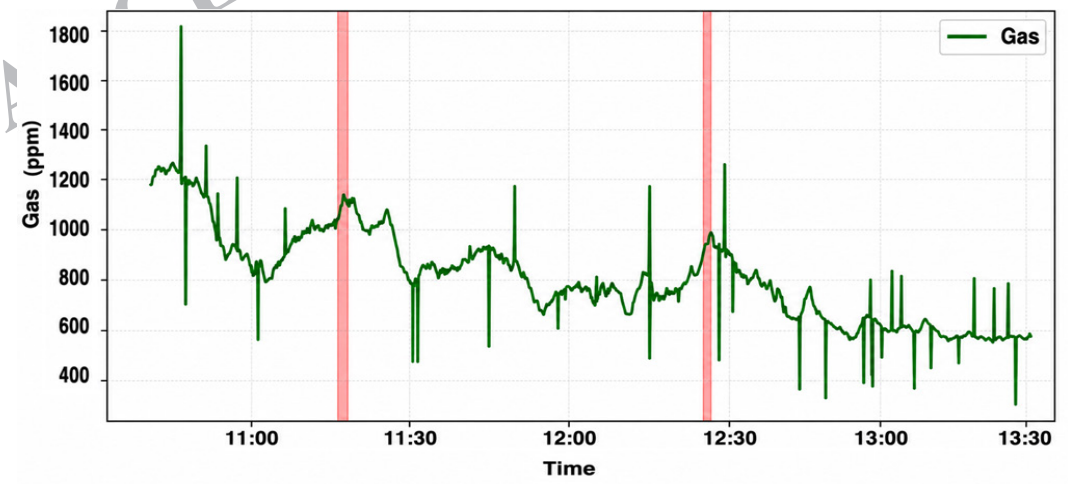

Fig. 8. Collective anomaly detection visualization in gas stream by OCSVM.

2.5.3. Combined anomaly

Combined anomalies are system-wide anomalies as they capture the interdependencies among all streams, allowing complex and system-level anomalous behavior detection. Combined anomaly detection visualized in Figure 9 shows that H-RADIS effectively identifies sharp deviations (e.g., 22:15, 22:30, 22:45) in HVAC streams to give a consistent and accurate picture of system

conditions. Unlike H-RADIS, LOF frequently identifies smaller deviations as anomalies, like data points around 23:22 in Figure 10 are marked as anomalies. On the same note, OCSVM identifies data points as anomalies, which are significant deviations at certain points, such as data points (23:21 and 23:25) in Figure 11 are marked as combined anomalies. As a whole, both LOF and OCSVM exhibit struggling behavior than H-RADIS in capturing combined anomalies that enable detection of broader systemic changes in the operational status of the HVAC system.

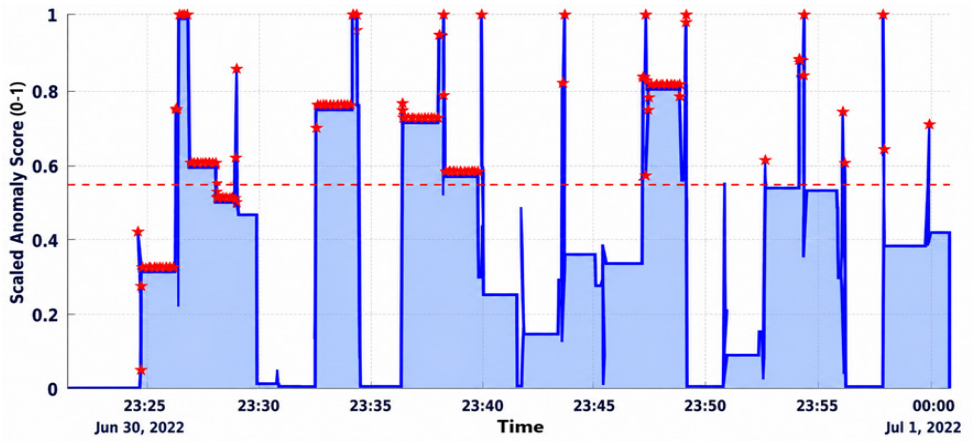

Fig. 9. Combined anomaly detection visualization in HVAC streams by H-RADIS.

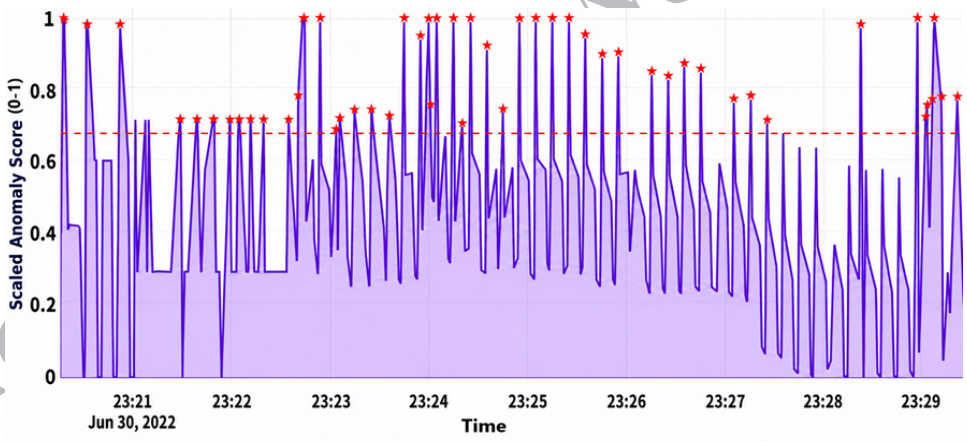

Fig. 10. Combined anomaly detection visualization in HVAC streams by LOF.

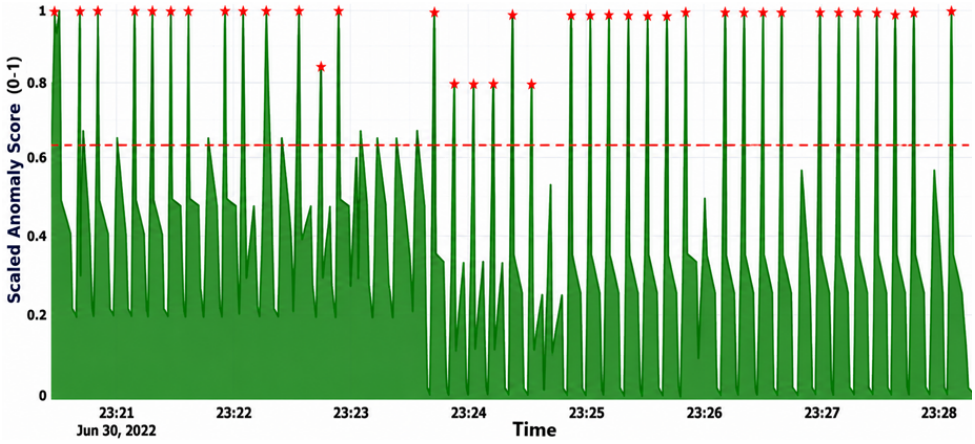

Fig. 11. Combined anomaly detection visualization in HVAC streams by OCSVM.

3. RESULTS AND DISCUSSION

Evaluation of the proposed framework, H-RADIS, is performed from two key perspectives: accurate detection of anomalies and optimal resource utilization across resource-constrained devices. To assess its suitability for deployment on resource-constrained devices, system-level metrics, including processing time, throughput, memory consumption, and energy usage, are used. Meanwhile, to evaluate the anomaly-detection efficiency of H-RADIS, the metrics accuracy, precision, recall, F1 score, and ROC-AUC are used. Moreover, robustness and consistency are analyzed through cross-dataset variation and stability measures, while statistical significance tests are also conducted to validate the reliability of the observed improvements. A detailed discussion on performance evaluation is given in the subsequent section.

3.1. Sensor-Level and System-Level Performance Analysis

The stream-wise comparative results for all sensors and models are given in Table 3. This comparison demonstrates that H-RADIS maintains a balanced performance across both datasets. For Temperature, Humidity, Gas, and Light sensors, H-RADIS always has better precision, recall, F1-score, and ROC-AUC trade-offs than LOF and OCSVM, indicating its capability to identify anomalies with fewer false positives by maintaining sensitivity. More importantly, the near-perfect scores by the H-RADIS, LOF, and OCSVM for the Accelerometer stream are mostly attributed to the nature of the underlying data, whereby the scores are characterized by low intra-class variance and more distinct between normal and abnormal patterns of data, which makes the classification task less complex. Nevertheless,

MANUSCRIPT

Table 3. Sensor-level performance analysis of H-RADIS, LOF, and OCSVM for dataset-1 and dataset-2.

Sensor	Model	Dataset-1					Dataset-2				
		Accuracy	Precision	Recall	F1-Score	ROC-AUC	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Temperature	H-RADIS	0.99	0.96	0.84	0.90	0.97	0.99	0.95	0.69	0.80	0.88
	LOF	0.95	0.51	0.67	0.58	0.95	0.99	0.90	0.78	0.83	0.94
	OCSVM	0.97	0.73	0.76	0.74	0.96	0.97	0.92	0.35	0.50	0.61
Humidity	H-RADIS	0.98	0.94	0.78	0.86	0.96	0.99	0.87	0.76	0.81	0.92
	LOF	0.93	0.37	0.46	0.41	0.81	0.99	0.84	0.79	0.81	0.95
	OCSVM	0.96	0.66	0.62	0.63	0.88	0.98	0.73	0.46	0.58	0.56
Gas	H-RADIS	0.96	0.60	0.52	0.56	0.85	0.98	0.78	0.75	0.77	0.91
	LOF	0.89	0.09	0.19	0.12	0.33	0.96	0.50	0.75	0.60	0.95
	OCSVM	0.93	0.35	0.36	0.34	0.59	0.98	0.95	0.50	0.66	0.58
Accelero-meter	H-RADIS	0.99	0.98	0.99	0.99	0.99	0.99	0.93	0.93	0.93	0.98
	LOF	0.99	0.95	0.83	0.91	0.94	0.99	0.92	0.76	0.87	0.93
	OCSVM	0.99	0.97	0.91	0.95	0.96	0.99	0.94	0.70	0.92	0.57
Light	H-RADIS	0.98	0.89	0.70	0.80	0.93	0.99	0.90	0.83	0.87	0.96
	LOF	0.96	0.64	0.67	0.66	0.94	0.97	0.60	0.77	0.67	0.96
	OCSVM	0.97	0.77	0.70	0.73	0.93	0.99	0.87	0.67	0.76	0.53

in this simplified environment, H-RADIS has a more balanced precision, recall, F1-score, and ROC-AUC. In general, the stream-wise analysis suggests that H-RADIS offers more trustworthy and generalizable anomaly detection results for heterogeneous sensor streams across all metrics.

3.2. Aggregated Performance Analysis

In order to obtain the overall performance of anomaly detection, the performance of each model was aggregated over the five data streams (temperature, humidity, gas, accelerometer, and light) and two datasets. For each performance metric: accuracy, precision, recall, F1-score, and ROC-AUC the 10 corresponding observations (five sensors × two datasets) were collected, and their arithmetic mean and standard deviation were calculated. The mean is used to represent overall performance of the detection model, and the standard deviation represents the variability of performance among different sensors and data. Therefore, the result of the aggregated analysis is represented as Mean ± SD. This process summarizes the performances over different types of sensors and datasets concisely while retaining the information of accuracy, precision, recall, F1-score, and ROC-AUC, which are useful for the anomaly detection problem. A more refined comparative analysis based on aggregated analysis given in Table 4 and shown in Figure 12, which represents the aggregated and consolidated performance of all models for both datasets. This type of evaluation method is similar to comprehensive time-series anomaly detection studies that do evaluations of the methods on a series of univariate and multivariate time-series datasets and then summarize the results to compute and report the overall effectiveness, efficiency, and robustness of the methods as a whole according to various data characteristics [26]. In this aggregated analysis, as shown in Figure 12, H-RADIS establishes itself as the most reliable model to achieve the highest overall performance across all the major evaluation metrics in comparison to its baseline models. H-RADIS achieves the highest Accuracy (0.987) overall, comparatively more than achieved by Ni *et al.* [16]. It also attains the best ROC-AUC (0.938) along with significant outcomes against Precision (0.880), Recall (0.786), and F1-score (0.829). As a whole, these results indicate better anomaly discrimination, better detection balance, and lower false positive capability of H-RADIS than the baseline models.

In the context of performance evaluation across two different datasets given in Table 4, a very small difference is observed, which means that H-RADIS exhibits consistent behavior across two evaluated datasets. In comparison, OCSVM exhibits moderate results but with a greater variation across both datasets, especially in Recall and F1-score, whereas LOF has relatively lowest Precision (0.634) and F1-score (0.623), indicating that it does not behave consistently across evaluated datasets. The aggregate analysis, on the whole, supports the fact that H-RADIS is not only better in terms of sensor-level performance but is also exhibits consistent behavior in aggregated analysis.

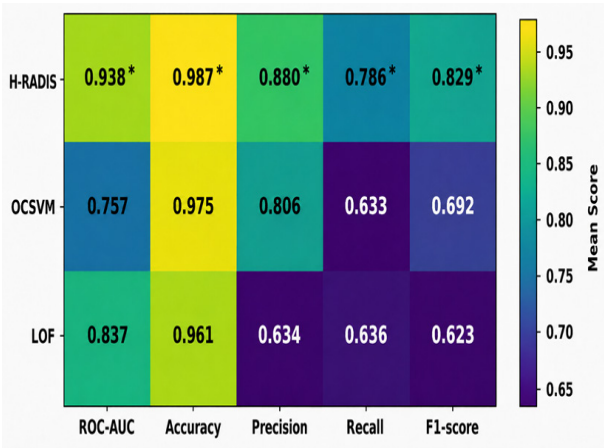

Fig. 12. Aggregated multi-metric performance comparison across all models and datasets.

Table 4. Aggregated performance analysis across all sensors and datasets.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
H-RADIS	0.98 ± 0.01	0.90 ± 0.14	0.78 ± 0.17	0.82 ± 0.15	0.94 ± 0.04
LOF	0.96 ± 0.03	0.59 ± 0.30	0.60 ± 0.23	0.60 ± 0.27	0.81 ± 0.19
OCSVM	0.97 ± 0.02	0.79 ± 0.20	0.60 ± 0.22	0.68 ± 0.22	0.75 ± 0.18

ACCEPTED MANUSCRIPT

3.3. Cross Dataset Robustness and Consistency Analysis

To assess H-RADIS consistent anomaly detection performance across different sensor streams and datasets, a paired cross-dataset variation analysis was performed. For each model, sensor, and evaluation metric, the absolute difference between the corresponding Dataset-1 and Dataset-2 performance values was calculated. The absolute differences were then averaged across the five sensor streams to obtain metric-specific cross-dataset variation for ROC-AUC, accuracy, precision, recall, and F1-score. The five metric-specific variations were subsequently averaged to obtain the overall average variation ($\bar{V}$). To facilitate interpretation in terms of stability, a study-defined Cross Dataset Consistency Index (CDCI) was calculated as CDCI $= 1 - \bar{V}$. Since the underlying performance metrics are normalized to the interval (0,1). The CDCI value closer to 1 indicates smaller average performance changes between datasets and shows greater cross-dataset consistency, whereas a lower value indicates greater sensitivity to dataset variation. The CDCI is used as a descriptive stability measure rather than as a conventional statistical reliability coefficient. This cross-dataset robustness and consistency analysis is based on prior anomaly detection studies that consider other factors like model stability, type of anomaly behavior, and consistent experimental protocols along with standard metrics to validate the model's performance [27].

Cross dataset robustness and consistency-driven comparison of the proposed and baseline models for both datasets is given in Table 5. In this comparative summary, H-RADIS reflects the best trade-off between high performance and low variability, which affirms that it has the least cross-dataset changes in performance across all key metrics. It also has the smallest average sensor variability, indicating that it is more consistent in its performance on heterogeneous sensing streams. In comparison, OCSVM and LOF have much greater shifts and variability, indicating that they are more susceptible to sensor-specific and dataset-specific variations. Finally, the highest value of consistency index CDCI indicates that H-RADIS demonstrates more consistent and reliable multi-sensor anomaly detection performance under the evaluated experimental conditions.

3.4. Statistical Significance Analysis

To complement the aggregated performance and cross-dataset consistency analysis, the statistical significance of the performance differences between H-RADIS and the baseline anomaly-detection methods, LOF and OCSVM, was evaluated using the Wilcoxon signed-rank test. This evaluation method is followed by previous anomaly detection task in which the Wilcoxon signed-rank test was applied to make the statistical comparison of multiple one-class classifiers [28].

This non-parametric paired test was selected because the performance measurements for H-RADIS and each competing model were obtained under corresponding experimental conditions and the test does not require the paired differences to follow a normal distribution. For each evaluation metric ROC-AUC, accuracy, precision, recall, and F1-score the corresponding H-RADIS and baseline results were treated as paired observations, and the Wilcoxon test examined whether the distribution of their paired differences was centered at zero. The null hypothesis (H_0) assumed no systematic difference between the paired performances of H-RADIS and the respective baseline, whereas the alternative hypothesis (H_1) indicated a systematic difference. Statistical significance was assessed at α=0.05; therefore, a p-value below 0.05 was

ACCEPTED MANUSCRIPT

Table 5. Cross-dataset variation and stability analysis on H-RADIS, LOF, and OCSVM.

Model	Cross-Dataset Variation					Average Sensor Variability	Consistency Index
	ROC-AUC	Accuracy	Precision	Recall	F1-Score		
H-RADIS	0.046	0.010	0.068	0.118	0.098	0.068	0.932
OCSVM	0.302	0.020	0.204	0.190	0.134	0.170	0.830
LOF	0.172	0.036	0.262	0.234	0.236	0.188	0.812

considered evidence against (H_0). The test provides inferential support for the descriptive findings obtained from the aggregated and consistency analysis, while non-significant results indicate that the available paired observations do not provide sufficient evidence of a systematic difference for that particular metric.

Table 6 presents the statistical significance analysis based on the Wilcoxon signed-rank test applied to the paired Dataset x Sensor observations. The results show that, compared with OCSVM, H-RADIS achieves statistically significant improvements across all anomaly-detection metrics except precision means that, the Wilcoxon signed-rank test did not detect a statistically significant difference in precision between H-RADIS and OCSVM. In comparison with LOF, H-RADIS demonstrates statistically significant improvements in accuracy, precision, recall, and F1-score, while the difference in ROC-AUC is not statistically significant. In general, statistical analysis demonstrates that the observed differences are statistically supported for most metrics under the evaluated experimental conditions. These results suggest that, under the evaluated sensor and dataset conditions, H-RADIS provides a comparatively balanced and robust anomaly-detection performance relative to the baseline models.

Table 6. Statistical significance analysis using the Wilcoxon Signed-Rank test.

Comparison	Metric	p-value	Significant
H-RADIS VS OCSVM	ROC-AUC	0.0098	Yes
	Accuracy	0.0156	Yes
	Precision	0.1758	No
	Recall	0.0020	Yes
	F1-score	0.0039	Yes
H-RADIS VS LOF	ROC-AUC	0.3008	No
	Accuracy	0.0039	Yes
	Precision	0.0117	Yes
	Recall	0.0195	Yes
	F1-score	0.0039	Yes

3.5. Computational Performance Analysis

The resource utilization efficiency of H-RADIS against baselines is measured using processing time, throughput, memory usage, energy consumption and CPU usage metrics. These metrics play a crucial role in the evaluation of streaming algorithms' performance for their deployment on edge devices. All runtime measurements were collected during execution of the online processing loop.

Processing time was measured as the elapsed wall-clock time using Python's high-resolution time. perf_counter() function as given in Equation (2).

$$T_{processing} = T_{end} \times T_{start} \quad (2)$$

Where T_{start} and T_{end} denote the start and end times of the online execution process, respectively.

Throughput was calculated as the number of valid sensor instances processed per unit processing time as given in Equation (3).

$$Throughput = \frac{N_{valid}}{T_{processing}} \quad (3)$$

Where N_{valid} represents the total number of valid sensor instances processed. Throughput is therefore reported in sensor operations per second.

CPU Processing Time was obtained through psutil as the change in the sum of user and system CPU times, and average CPU utilization was computed using the formula given in Equation (4).

$$CPU\ Utilization\ (\%) = \frac{T_{CPU}}{T_{processing}} \times 100 \quad (4)$$

Process memory consumption was monitored using the resident set size (RSS), and the maximum sampled value was reported as peak RAM usage by following the calculations in Equation (5).

$$Peak\ RAM\ (MB) = \frac{max(RSS)}{1024^2} \quad (5)$$

Where percentage-based memory utilization is required, it may be calculated as given in Equation (6).

$$Memory\ Utilization\ (\%) = \frac{Process\ RSS}{Total\ System\ Memory} \times 100 \quad (6)$$

However, in our implementation, memory consumption is reported as peak RAM usage in megabytes.

ACCEPTED MANUSCRIPT

Energy consumption was analytically estimated using the assumed processor power and measured CPU execution time as given in the following Equation (7).

$$E_{estimated} = P_{assumed} \times T_{CPU} \quad (7)$$

Where $P_{assumed}$ = 15 W.

Accordingly, processing time, throughput, CPU utilization, and peak RAM were derived from runtime observations, whereas energy consumption represents a model-based estimate rather than a direct hardware measurement. The comparative results on resource utilization by all models across both datasets and all metrics are given in Table 7.

It is noticeable that H-RADIS consistently performs well across both baseline models in terms of computational efficiency on both datasets. It achieves the highest throughput of 3,798.29 ops/s on Dataset-1 and 3,779.67 ops/s on Dataset-2 along with the lowest processing times (32.21 s and 32.96 s, respectively). Furthermore, H-RADIS achieves lowest energy consumption (483.00 J and 494.40 J) across dataset-1 and dataset-2 respectively, aligning with the energy-efficient characteristics of the Hybrid Fog-Edge architecture proposed by Islam *et al.* [15]. While the memory usage remains similar for all models (around 155-168 MB). The CPU usage across each model is also similar (around 99.9%), indicating that the workload for each method is equally challenging.

Moreover, to quantify the relative performance of H-RADIS against the baseline methods in terms of anomaly-detection effectiveness and computational efficiency, an improvement analysis was conducted. For each metric *m*, the mean performance of LOF and OCSVM was used to construct a collective baseline as given in the following Equation (8).

$$Bm = \frac{M_{LOF,m} + M_{OCSVM,m}}{2} \quad (8)$$

Where B_m denotes the collective baseline for metric *m*, while $M_{LOF,m}$ and $M_{OCSVM,m}$ represent the corresponding mean metric values of LOF and OCSVM, respectively.

For metrics where higher values indicate better performance, including accuracy, precision, recall, F1-score, ROC-AUC, and throughput, the relative gain was calculated using following Equation (9).

$$Gm = \frac{M_{H-RADIS,m} + B_m}{B_m} \times 100 \quad (9)$$

Where G_m denotes the relative gain (%) for metric *m* and $M_{H\text{-}RADIS,m}$ represents the corresponding metric value of H-RADIS.

For computational metrics where lower values indicate better performance, including processing time, memory usage, and energy consumption, the relative reduction was calculated using following Equation (10).

$$Rm = \frac{B_M - M_{H-RADIS,m}}{B_m} \times 100 \quad (10)$$

Where R_m denotes the relative reduction (%) for metric *m.* Positive values of G_m and R_m therefore indicate improvement by H-RADIS over the collective LOF-OCSVM baseline, with the direction of desirability explicitly considered for each metric.

ACCEPTED MANUSCRIPT

Table 7. Computational performance analysis of H-RADIS, LOF, and OCSVM.

Metric	Dataset-1			Dataset-2		
	H-RADIS	**LOF**	**OCSVM**	**H-RADIS**	**LOF**	**OCSVM**
Processing Time (s)	32.21	54.63	42.85	32.96	58.41	43.73
Throughput (ops/s)	3798.29	2284.51	2914.73	3779.67	2138.17	2858.22
Memory Usage (MB)	167.90	168.65	168.52	155.69	155.60	155.77
Energy Consumption (J)	483.00	818.70	642.75	494.40	876.00	655.65
CPU Usage (%)	99.96	99.91	99.99	99.98	99.95	99.98

For anomaly-detection performance, the metric-wise results based on Equation (9) are reported in Table 8 shows that H-RADIS improves all five-evaluation metrics over the collective LOF–OCSVM baseline. H-RADIS achieved relative gains of 26.08% in F1-score, 23.88% in recall, 22.22% in precision, 17.69% in ROC-AUC, and 1.96% in accuracy over the collective LOF-OCSVM baseline.

For the computational improvement, Table 9 follows the same baseline construction as defined in Equation (8). Relative gain in throughput was calculated using Equation (9), while reductions in processing time, memory usage, and energy consumption were calculated using Equation (10). Based on these calculations, H-RADIS achieved a 48.65% increase in throughput, together with reductions of 34.71% in processing time, 34.69% in energy consumption, and 0.21% in memory usage.

As a whole, the overall assessment through different analytical dimensions indicates the effectiveness of the proposed H-RADIS framework. The stream-level analysis shows that it is capable of sustaining a balanced trade-off between precision, recall, F1-score, and ROC-AUC among various sensor types. The aggregated analysis further supports its better average performance with minor variability over different datasets. These findings are further strengthening by the robustness and consistency analysis that indicates that H-RADIS has low cross-dataset variation and maintains a consistent behavior than LOF and OCSVM. Moreover, the statistical significance analysis indicates that several observed performance differences between H-RADIS and the baseline approaches are statistically supported within the evaluated experimental conditions. In terms of optimal resource utilization, H-RADIS has higher throughput, lower processing time, energy consumption and energy consumption with equal CPU usage, that are encouraging for real-time deployment. Lastly, the improvement analysis further indicates that H-RADIS achieved favorable gains over the collective LOF and OCSVM baseline across the evaluated detection and computational measures under the tested experimental conditions. Collectively, these findings indicate that, within the evaluated experimental conditions, H-RADIS

ACCEPTED MANUSCRIPT

Table 8. Anomaly detection improvement analysis of H-RADIS over LOF and OCSVM across dataset-1 and dataset-2.

Metric	H-RADIS Mean	LOF Mean	OCSVM Mean	Collective Baseline	Relative Improvement
ROC-AUC ↑	0.938	0.837	0.757	0.7970	**17.69%** ↑
Accuracy ↑	0.987	0.961	0.975	0.9680	**1.96%** ↑
Precision ↑	0.880	0.634	0.806	0.7200	**22.22%** ↑
Recall ↑	0.786	0.636	0.633	0.6345	**23.88%** ↑
F1-score ↑	0.829	0.623	0.692	0.6575	**26.08%** ↑

Table 9. Computational improvement analysis of H-RADIS over LOF and OCSVM across dataset-1 and dataset-2.

Metric	H-RADIS Mean	LOF Mean	OCSVM Mean	Collective Baseline	Relative Improvement
Throughput (ops/s) ↑	3788.980	2211.340	2886.475	2548.908	48.65% ↑
Processing Time (s) ↓	32.585	56.520	43.290	49.905	34.71% ↓
Memory Usage (MB) ↓	161.795	162.125	162.145	162.135	0.21% ↓
Energy Consumption (J) ↓	488.700	847.350	649.200	748.275	34.69% ↓

exhibits consistent detection performance and favorable computational efficiency for multi-type anomaly detection in heterogeneous streaming IoT environments.

4. CONCLUSION AND FUTURE WORK

This study presents H-RADIS, a novel real-time anomaly detection approach specifically designed for resource-constrained edge devices. Built upon the HST algorithm with certain variations of sliding window-based dynamic threshold with adaptive parameter tuning and dynamic scaling, H-RADIS is engineered for multi-stream anomaly detection, prioritizing the lowest resource utilization with significant accuracy, precision, and recall. In addition to integrating dynamic thresholds with adaptive parameter tuning, H-RADIS uses the Complex Event Processing (CEP) technique to detect collective anomalies (consecutive anomaly patterns) more effectively. The performance of H-RADIS is evaluated on two different datasets and compared in terms of anomaly detection accuracy and resource utilization efficiency with RIVER-based online machine learning algorithms, LOF, and OCSVM. The experimental results indicate that H-RADIS achieves significant relative improvement in anomaly detection with substantial gains in F1-score (26.08%), Recall (23.88%), Precision (22.22%), and ROC-AUC (17.69%). In resource utilization, H-RADIS also achieves resource efficiency gain contributed by 48.7% throughput improvement and significant reductions in processing time (34.70%), energy consumption (34.68%), and memory utilization (0.21%) with almost equal CPU utilization, considering H-RADIS a resource-efficient system under the evaluated experimental conditions.

Despite these significant outcomes, the applicability of this study is restricted by the use of a relatively small number of datasets and synthetically generated anomalies, which may impact the generalizability of the findings to broader real-world IoT environments. Future work will involve the validation of H-RADIS with real-world anomalies rather than synthetic anomalies, scalability to larger and more diverse IoT deployments, benchmarking against a wider variety of streaming anomaly-detection techniques, and integration with learning-based techniques to further enhance adaptability and detection accuracy.

5. ACKNOWLEDGEMENT

The authors acknowledge academic and research support from the Department of Computer Science, University of Sargodha, for the completion of this research task.

6. ETHICAL STATEMENT

This study did not involve human participants or animals; therefore, ethical approval was not required.

7. CONFLICT OF INTEREST

The authors have no conflict of interest regarding this research task.

8. FUNDING

No funding was received to conduct this research.

9. AUTHORSHIP CONTRIBUTION STATEMENT

Mubashir Abbas contributed to the idea proposal, literature review, research execution, and implementation. Humaira Ijaz designed the study, provided guidelines, and made final approval of the draft. The methodology development and experiments were carried out by Muhammad Fahad, and Bushra Jamil participated in comparative analysis, visualization, and validation. While Bilal Ashraf helped with testbed setup, experiments, and analysis, and technical proofreading was done by Kashif Munir.

10. DECLARATION

The authors affirm that the results given in this paper are original and the content has not been published before and is not under consideration for publication elsewhere. The manuscript has been read and accepted for publication by all the authors, and each has approved the final version. The authors also agree to transfer the copyright of the article to the Pakistan Academy of Sciences after the article's acceptance for publication. For simulation purpose, a publicly available smart building dataset is used, which is properly cited to avoid copyright infringement.

11. REFERENCES

1. A. Alawad and V. Marttin. Internet of Things (IoT) and Smart Applications. In: Pioneer and Innovative

ACCEPTED MANUSCRIPT

Studies in Computer Sciences and Engineering. A. Alaybeyoğlu (Ed.). *All Sciences Academy, Konya, Turkey* pp. 126-141 (2024).

2. N.M. Quy, L.A. Ngoc, N.T. Ban, N.V. Hau, and V.K. Quy. Edge computing for real-time internet of things applications: Future internet revolution. *Wireless Personal Communications* 132(2): 1423-1452 (2023). https://doi.org/10.1007/s11277-023-10669-w
3. C. Hu, Z. Sun, C. Li, Y. Zhang, and C. Xing. Survey of Time Series Data Generation in IoT. *Sensors* 23(15): 6976 (2023). https://doi.org/10.3390/s23156976
4. K.-D. Kang. A Review of Efficient Real-Time Decision Making in the Internet of Things. *Technologies* 10(1): 12 (2022). https://doi.org/10.3390/technologies10010012
5. W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang. A Survey on the Edge Computing for the Internet of Things. *IEEE Access* 6: 6900–6919 (2018). https://doi.org/10.1109/ACCESS.2017.2778504
6. M. J. C. S. Reis and C. Serôdio. Edge AI for Real-Time Anomaly Detection in Smart Homes. *Future Internet* 17(4): 179 (2025). https://doi.org/10.3390/fi17040179
7. L. Erhan, M. Ndubuaku, M. Di Mauro, W. Song, M. Chen, G. Fortino, O. Bagdasar, and A. Liotta. Smart anomaly detection in sensor systems: A multi-perspective review. *Information Fusion* 67: 64-79 (2020). https://doi.org/10.1016/j.inffus.2020.10.001
8. T. Lu, L. Wang, and X. Zhao. Review of anomaly detection algorithms for data streams. *Applied Sciences* 13(10): 6353 (2023). https://doi.org/10.3390/app13106353
9. Y. Zhang, C. Jiang, B. Yue, J. Wan, and M. Guizani. Information fusion for edge intelligence: A survey. *Information Fusion* 81: 171-186 (2022). https://doi.org/10.1016/j.inffus.2021.11.018
10. A. Verdone, S. Scardapane, and M. Panella. A data attribution approach for unsupervised anomaly detection on multivariate time series. *Expert Systems with Applications* 297: 129285 (2026). https://doi.org/10.1016/j.eswa.2025.129285
11. Z. Zhong, Z. Yu, Y. Yang, W. Wang, K. Yang, and C. Chen. PatchAD: A lightweight patch-based MLP-mixer for time series anomaly detection. *IEEE Transactions on Big Data* 11(6): 3460-3473 (2025). https://doi.org/10.1109/TBDATA.2025.3596745
12. A. Ayad, M. El-Gayar, N. Hikal, and N. Sakr. Efficient real-time anomaly detection in IoT networks using one-class autoencoder and deep neural network. *Electronics* 14(1): 104 (2025). https://doi.org/10.3390/electronics14010104
13. X. Wang, Z. Tang, J. Guo, T. Meng, C. Wang, T. Wang, and W. Jia. Empowering edge intelligence: A comprehensive survey on on-device AI models. *ACM Computing Surveys* 57(9): 228 (2025). https://doi.org/10.1145/3724420
14. V. Prakash, O. Odedina, A. Kumar, L. Garg, and S. Bawa. A secure framework for the Internet of Things anomalies using machine learning. *Discover Internet of Things* 4(1): 33 (2024). https://doi.org/10.1007/s43926-024-00088-z
15. U. Islam, M. Alatawi, A. Alqazzaz, S. Alamro, B. Shah, and F. Moreira. A hybrid fog-edge computing architecture for real-time health monitoring in IoMT systems with optimized latency and threat resilience. *Scientific Reports* 15(1): 25655 (2025). https://doi.org/10.1038/s41598-025-09696-3
16. C. Ni, J. Wu, and H. Wang. Energy-aware edge computing optimization for real-time anomaly detection in IoT networks. *Applied and Computational Engineering* 139: 42-53 (2025). https://doi.org/10.54254/2755-2721/2025.22280
17. Z. Chen, M. Wang, J. Li, G. Li, and X. Wu. Unsupervised anomaly detection methods for in situ observation data: A comparative study. *IEEE Access* 13: 187347-187360 (2025). https://doi.org/10.1109/ACCESS.2025.3626371
18. N. Watanabe, T. Yamazaki, T. Miyoshi, R. Yamamoto, M. Nakahara, N. Okui, and A. Kubota. Self-adaptive traffic anomaly detection system for IoT smart home environments. *IEICE Transactions on Communications* 108(3): 230-242 (2025). https://doi.org/10.23919/transcom.2024EBT0002
19. J. Montiel, M. Halford, S.M. Mastelini, G. Bolmier, R. Sourty, and *et al*. RIVER: Machine learning for streaming data in Python. *Journal of Machine Learning Research* 22: 1-8 (2021). http://jmlr.org/papers/v22/20-1380.html
20. S. Tan, K. Ting, and T. Liu. Fast anomaly detection for streaming data. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence* (*IJCAI)* pp. 1511-1516 (2011). https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-254
21. Z. Cheng, C. Zou, and J. Dong. Outlier Detection using Isolation Forest and Local Outlier Factor. *Conference on Research in Adaptive and Convergent Systems (RACS '19), (24th - 27th September 2019), Chongqing, China,* pp. 161-168 (2019). https://doi.org/10.1145/3338840.3355641
22. K. Yang, S. Kpotufe, and N. Feamster. An efficient one-class SVM for anomaly detection in the Internet

ACCEPTED MANUSCRIPT

of Things. *arXiv preprint arXiv:2104.11146* (2021). https://doi.org/10.48550/arXiv.2104.11146

23. S.K. Ray and S. Susan. Performance analysis of online machine learning frameworks for anomaly detection in IoT data streams. *15th International Conference on Computing, Communication and Networking Technologies (ICCCNT), (24th -28th June 2024), Kamand, India, (2024)*. https://doi.org/10.1109/ICCCNT61001.2024.10724326
24. U. Erol, F. Raimondo, J. Pope, S. Gunner, V. Kumar, and *et al*. Multi-sensor, multi-device smart building indoor environmental dataset. *Data in Brief* 49: 109392 (2023). https://doi.org/10.1016/j.dib.2023.109392
25. H. Kim and I. Joe. Enhancing anomaly detection in maritime operational IoT time-series data with synthetic outliers. *Electronics* 13(19): 3912 (2024). https://doi.org/10.3390/electronics13193912
26. S. Schmidl, P. Wenig, and T. Papenbrock. Anomaly detection in time series: A comprehensive evaluation. *Proceedings of the VLDB Endowment* 15(9): 1779-1797 (2022). https://doi.org/10.14778/3538598.3538602
27. N. Mejri, L. Lopez-Fuentes, K. Roy, P. Chernakov, E. Ghorbel, and D. Aouada. Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods. *Expert Systems with Applications* 256: 124922 (2024). https://doi.org/10.1016/j.eswa.2024.124922
28. W. Fernandes Jr., K.S. Komati, and K.A.S. Gazolli. Anomaly detection in oil-producing wells: A comparative study of one-class classifiers in a multivariate time series dataset. *Journal of Petroleum Exploration and Production Technology* 14: 343-363 (2024). https://doi.org/10.1007/s13202-023-01710-6

ACCEPTED MANUSCRIPT