CLUSTER-LLM: CLUSTERING-GUIDED LLM-BASED DATA AUGMENTATION FOR ANOMALY DETECTION IN SMART HOME

Anonymous authors
Paper under double-blind review

ABSTRACT

Anomaly detection in activities of daily living (ADL) within smart home environments is critical for supporting the safety and well-being of older adults living independently. However, the scarcity of annotated anomaly data and the inherent behavioral heterogeneity among residents pose significant challenges to developing robust detection models. To address these issues, we propose a novel clustering-guided LLM-based data augmentation framework (Cluster-LLM) for ADL anomaly detection. This framework first leverages clustering algorithms to mine differential behavioral patterns from raw sensor data, and then generates lifelike synthetic activity sequences matching the characteristics of each behavioral group through LLM reasoning. We conduct extensive evaluations on four clustering approaches, namely K-means, K-Medoids, DBSCAN and deep embedded clustering (DEC), using two standard smart home datasets. Extensive experiments with four one-class classifiers and seven evaluation metrics demonstrate that Cluster-LLM achieves satisfactory performance in anomaly detection in smart home. Specifically, Cluster-LLM boosts the best F1-score on the Kyoto7 dataset from 0.444 to 0.667, representing a 50.2% relative improvement over baselines. On the unsupervised benchmark, the proposed framework enhances accuracy across diverse clustering backbones, while maintaining robust F1 gains within a moderate augmentation scale.

1 INTRODUCTION

Ambient sensor-based Abnormal behavior detection (ABD) in smart homes supports older adults' independent living by continuously monitoring Activities of Daily Living (ADL)—cooking, hygiene, sleeping—and flagging deviations Chandola et al. (2009); Achirei & Alboaie (2024); Zhang et al. (2023). Since ADL reflects cognitive and physical capabilities, ABD serves as an early indicator of health deterioration or safety hazards Roy et al. (2016); Anikwe et al. (2022); Ohlan & Ohlan (2022).

Existing ABD paradigms primarily encompass supervised, semi-supervised, and unsupervised one-class learning approaches Tong et al. (2015); Tay et al. (2023); Nawaz et al. (2025). All three paradigms suffer from an inherent bottleneck: *training data scarcity and narrow behavioral coverage*. Because abnormal events are naturally rare and acquiring fine-grained, long-term annotations incurs prohibitive labor costs and privacy concerns Prenkaj et al. (2023); Tay et al. (2023), supervised models suffer from catastrophic class imbalance, while unsupervised one-class classifiers frequently fail to bound the full spectrum of normal and abnormal human behaviors Dhekane & Plötz (2025).

Prior work has tackled data scarcity through two routes. Traditional augmentation—SMOTE Chawla et al. (2002), ADASYN He et al. (2008), and perturbation—interpolates or corrupts existing samples but cannot preserve temporal dependencies in sensor sequences or introduce novel behavioral patterns. More recently, LLMs have become generative tools for smart-home data. SHADE-AD Fu et al. (2025) prompts an LLM to produce anomalous sequences for class balance; AgentSense Leng et al. (2025) deploys multiple LLM-based personas across 22 homes. These methods still rely on static prompts or agent profiles not grounded in the empirical behavioral clusters of real sensor streams. When synthetic samples drift from the true resident distribution, classifiers learn skewed boundaries that generalize poorly. Without conditioning on observed subgroups, generated sequences cluster around a single "average" routine, diluting the boundary diversity that anomaly detectors need.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these limitations, we present **Cluster-LLM**, a clustering-conditioned LLM augmentation framework that generates diverse normal and anomalous ADL sequences to train anomaly detectors. Rather than treating residents as a single population, Cluster-LLM first discovers behavioral subgroups through unsupervised clustering, then conditions LLM generation on the statistical profile of each subgroup. This two-stage design anchors synthetic data to empirically observed behavioral patterns, expanding the diversity of both normal routines and anomalous deviations that classifiers encounter during training. We instantiate this idea across four clustering paradigms, including centroid-based (K-means), medoid-based (K-Medoids), density-based (DBSCAN), and deep embedded (DEC), to understand how different representations of behavioral similarity affect generation quality. Moreover, we evaluate our framework on two public smart-home benchmarks: Kyoto7 and HH111, where the former provides a scripted testbed with injected errors for supervised anomaly detection, and HH111 delivers long-term, unscripted sensor streams with rule-derived labels for unsupervised settings. These benchmark setup provides a comprehensive stress test for cross-paradigm generalization. Results show that Cluster-LLM consistently outperforms non-augmented baselines. The contributions are summarized as follows:

- We propose Cluster-LLM, the first clustering-conditioned data augmentation framework for sensor-based ADL anomaly detection. Unlike prior LLM generators that rely on a single global prompt, Cluster-LLM builds cluster-specific behavioral profiles to guide LLM sequence generation. This conditioning mitigates synthetic data homogeneity and diversifies the training distribution under severe class imbalance.
- We formulate a unified conditioning pipeline that integrates structural, density-based, and deep latent behavioral representations into structured LLM prompts, providing a rigorous blueprint for bridging unsupervised pattern discovery with generative augmentation.
- Extensive experiments across two public smart home benchmarks, validate that Cluster-LLM consistently achieves superior AUC and F1 performance than baselines. We also delineate the regime where cluster conditioning is effective, revealing a non-monotonic scaling behavior with synthetic data volume, which yields guidelines for balancing real and synthetic data, effectively mitigating distribution shift during deployment.

2 Related Work

Sensor-Based ADL Anomaly Detection. Research on sensor-based Activity of Daily Living (ADL) analysis spans three related but distinct directions: activity recognition, anomaly detection, and behavior change detection. ADL recognition focuses on classifying sensor sequences into predefined activity categories such as cooking or hygiene Chernbumroong et al. (2013); Ramirez & Cardinale (2022). Behavior change detection monitors gradual deviations in routine patterns to flag early signs of cognitive or physical decline Prenkaj et al. (2023); Riboni et al. (2016). Our work focuses on *anomaly detection*, which aims to identify rare, acute deviations that indicate immediate safety hazards or health emergencies.

Within anomaly detection, One-Class Classification (OCC) has become the dominant paradigm because smart home streams are overwhelmingly normal Jakkula & Cook (2011); Yahaya et al. (2021b). Early work by Jakkula and Cook Jakkula & Cook (2011) showed that OCSVM can learn normal behavioral boundaries from unlabeled sensor sequences. Subsequent research strengthened robustness through ensemble stacking Yahaya et al. (2021a), Hidden Conditional Random Fields for temporal dependencies Tong et al. (2015), and density-based outlier methods Yahaya et al. (2019). Despite these algorithmic advances, all OCC models remain constrained by how well training data covers the resident's behavioral repertoire. When normal training sequences miss rare but legitimate routines, classifiers produce excessive false alarms Yahaya et al. (2021b).

Synthetic Smart Home Behavior and Sensor Generation. Early work tackled class imbalance through feature-space interpolation. SMOTE Chawla et al. (2002) and ADASYN He et al. (2008) synthesize minority-class samples by interpolating between nearest neighbors. VAE-based approaches such as MGVAE Ai et al. (2023) learn generative latent distributions to produce new examples. Perturbation methods add noise or swap attributes. These techniques share a common weakness: they operate in feature space and rarely preserve the temporal logic of sensor event sequences.In the smart home domain, synthetic data has traditionally come from rule-based simulators.

More recently, LLMs have been explored for smart-home data generation. ADLGen You et al. (2026) uses structured symbolic templates with temporal constraints to synthesize rare activity traces. AgentSense Leng et al. (2025) scales this idea by deploying multiple LLM agents, each embodying a distinct persona, to generate diverse resident behaviors across 22 simulated homes. SHADE-AD Fu et al. (2025) focuses on clinical settings, explicitly prompting an LLM to generate anomalous sequences for Alzheimer's patients. These methods improve diversity, yet they still rely on manually designed prompts or agent profiles that are not derived from the empirical clusters present in real sensor data. Without grounding generation in observed behavioral subgroups, synthetic samples may drift from true resident distributions and weaken downstream detection.

Clustering for Behavioral Modeling in Smart Homes. Clustering has long been used to uncover latent behavioral patterns in sensor data. In ADL recognition, K-Means and K-Medoids partition feature spaces to discover routine categories Lloyd (1982); Kaufman & Rousseeuw (1987), while DBSCAN identifies irregular behavior by flagging low-density instances as noise Ester et al. (1996). In anomaly detection, these same techniques help define normal behavioral boundaries: dense clusters represent typical routines, and points that fall outside them signal potential anomalies Yahaya et al. (2021b). More recently, Deep Embedded Clustering Xie et al. (2016) has been applied to learn non-linear sensor feature representations before grouping, capturing complex temporal patterns that raw features miss Ren et al. (2025). Despite this progress, clustering has served mainly as an analysis tool—discovering groups, not generating data. No prior work uses empirical cluster profiles to condition LLM-based synthesis of ADL sequences. Cluster-LLM fills this gap by converting discovered behavioral subgroups into structured generation prompts, so that synthetic data explicitly reflects the diversity observed in real sensor streams.

3 PRELIMINARIES

3.1 PROBLEM FORMULATION

Consider a smart-home environment equipped with a spatial sensor network $\mathbf{S} = \{s_1, s_2, \ldots, s_M\}$ deployed across functional spaces (e.g., kitchen, living room, bedroom, bathroom, entrance). Each sensor $s \in \mathbf{S}$ triggers a discrete state change event represented as a tuple $e = (t, s, m)$, where $t \in \mathbb{R}^+$ denotes the continuous timestamp, $s \in \mathbf{S}$ is the sensor identifier, and $m \in \mathcal{M}$ represents the status message (e.g., `ON`/`OFF`). An illustration of this environment and the data processing pipeline appears in Figure A1(a) .

An *activity instance* is a temporal sequence of sensor events $\xi = [e_1, e_2, \ldots, e_L]$ collected within a fixed window, as shown in Figure A1(b). A feature extractor $\phi(\cdot)$ maps each sequence to a vector $x = \phi(\xi) \in \mathbb{R}^d$ that encodes spatiotemporal and transition statistics. The full real-world dataset is $\mathcal{D}_{\text{real}} = \{(x_i, y_i)\}_{i=1}^{N}$, where $y_i \in \{0, 1\}$ indicates normal (0) or anomalous (1) labels when available. We partition $\mathcal{D}_{\text{real}}$ into a training set $\mathcal{D}_{\text{train}}$ and a held-out test set $\mathcal{D}_{\text{test}}$; the test set always contains real data only. In the *known-anomaly* setting (Kyoto7), training instances carry labels that guide synthetic anomaly injection; in the *unsupervised* setting (HH111), training uses only normal sequences, and labels are reserved for evaluation. Due to the scarcity of abnormal events, $\mathcal{D}_{\text{train}}$ is constrained in behavioral diversity. We synthesize $\mathcal{D}_{\text{syn}} = \{x_j^{\text{syn}}\}_{j=1}^{N_{\text{syn}}}$ to broaden behavioral coverage without distorting the true distribution. The augmented set is $\mathcal{D}_{\text{aug}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{syn}}$, on which we train a detector $f : \mathbb{R}^d \to \{0, 1\}$ that generalizes to $\mathcal{D}_{\text{test}}$. All symbols are in Table A2.

3.2 CLUSTERING APPROACHES

To discover implicit sub-population behavior manifolds for prompt conditioning, we formalize four representative clustering paradigms over the feature space.

K-means-based Clustering. It partitions the training feature set $\mathcal{D}_{\text{train}} = \{\mathbf{v}_i\}_{i=1}^{N} \subset \mathbb{R}^d$ into K non-overlapping clusters $\mathcal{G} = \{G_1, \ldots, G_K\}$ by minimizing the intra-cluster variance:

$$\min_{\{G_1,\ldots,G_K\}} \sum_{k=1}^{K} \sum_{\mathbf{v} \in G_k} \|\mathbf{v} - \boldsymbol{\mu}_k\|_2^2, \tag{1}$$

where $\boldsymbol{\mu}_k = \frac{1}{|G_k|} \sum_{\mathbf{v} \in G_k} \mathbf{v}$, denotes the mean centroid of cluster G_k Lloyd (1982).

K-Medoids-based Clustering. K-Medoids enforces cluster representatives to be actual observed data points rather than virtual computed means. The objective function is:

$$\min_{\{\mathbf{m}_1,\ldots,\mathbf{m}_K\}\subset\mathcal{D}_{\text{train}}} \sum_{k=1}^{K}\sum_{\mathbf{v}\in G_k} d(\mathbf{v},\mathbf{m}_k), \tag{2}$$

where $\mathbf{m}_k \in \mathcal{D}_{\text{train}}$ is the exemplar medoid of cluster G_k, and $d(\cdot,\cdot)$ is an arbitrary pairwise distance metric Kaufman & Rousseeuw (1987); Schubert & Rousseeuw (2021). K-Medoids offers dual advantages: (1) medoids correspond to physically grounded behavioral instances $\mathbf{v} \in \mathcal{D}_{\text{train}}$, serving as concrete in-context examples for LLM reasoning; and (2) it natively supports non-Euclidean domain-specific metrics.

DBSCAN-based Clustering. DBSCAN groups behavioral instances based on local density without a predefined cluster count K Ester et al. (1996). For a feature vector $\mathbf{v}$, its ϵ-neighborhood is defined as $N_\epsilon(\mathbf{v}) = \{\mathbf{v}' \in \mathcal{D}_{\text{train}} \mid d(\mathbf{v},\mathbf{v}') \le \epsilon\}$. A point $\mathbf{v}$ is designated as a core point if $|N_\epsilon(\mathbf{v})| \ge \text{MinPts}$, where MinPts sets the density threshold. Instances residing in low-density regions are isolated as noise points $\mathcal{O}_{\text{noise}}$, which ground the LLM regarding atypical behavioral patterns.

DEC-based Clustering. Deep Embedded Clustering (DEC) jointly optimizes a non-linear mapping $f_\phi : \mathbb{R}^d \to \mathcal{Z} \subset \mathbb{R}^{d'}$ into a low-dimensional latent space and performs soft cluster assignment Xie et al. (2016). The probability q_{ik} of assigning latent representation $\mathbf{z}_i = f_\phi(\mathbf{v}_i)$ to cluster center $\boldsymbol{\mu}_k$ is modeled via Student's t-distribution:

$$q_{ik} = \frac{(1 + \|\mathbf{z}_i - \boldsymbol{\mu}_k\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'}(1 + \|\mathbf{z}_i - \boldsymbol{\mu}_{k'}\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{3}$$

where α denotes the degrees of freedom. DEC refines representations by minimizing the Kullback-Leibler (KL) divergence between $Q = [q_{ik}]$ and an auxiliary target distribution $P = [p_{ik}]$:

$$\mathcal{L}_{\text{DEC}} = KL(P \parallel Q) = \sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik} \log \frac{p_{ik}}{q_{ik}}, \tag{4}$$

where the target distribution P sharpens the soft assignments by squaring and renormalizing against the cluster marginals: $p_{ik} = \left(q_{ik}^2/f_k\right) \big/ \sum_{j=1}^{K} q_{ij}^2/f_j$, with $f_k = \sum_{i=1}^{N} q_{ik}$ the marginal mass of cluster k. DEC is uniquely suited for untangling non-linear spatial-temporal dependencies in high-dimensional sensor data.

4 METHODOLOGY

This section introduces the Cluster-LLM framework as illustrated in Figure 1, establishing a unified pipeline that covers the clustering scheme, LLM-based generation process, and data merging strategy. Given the training set $\mathcal{D}_{\text{train}}$, the pipeline works through three phases: behavioral clustering partitions $\mathcal{D}_{\text{train}}$ into homogeneous subgroups, LLM-based reasoning generates cluster-specific synthetic sequences, and data merging merges real and synthetic data into the augmented dataset $\mathcal{D}_{\text{aug}}$ that trains anomaly detectors.

4.1 CLUSTERING-BASED DATA PARTITIONING

Clustering-based data partitioning divides complex behavioral streams into coherent subgroups to provide structured context for LLM sequence synthesis. Formally, the training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ serves as input, where each $x_i \in \mathbb{R}^d$ is a feature vector extracted from sensor event sequences by $\phi(\cdot)$, and $y_i \in \{0, 1\}$ is the anomaly label. Residents differ in daily routines—some spend long stretches in the living room, others move frequently between kitchen and bathroom. To capture this heterogeneity, we partition $\mathcal{D}_{\text{train}}$ into K subgroups using the four clustering methods defined in Section 3.2 (K-Means, K-Medoids, DBSCAN, DEC). Each method produces a partition $\mathcal{G} = \{G_1, \ldots, G_K\}$ with $G_k \subset \mathcal{D}_{\text{train}}$, grouping instances whose feature vectors are close in the d-dimensional space. For every cluster G_k, we compute a behavioral profile $\mathcal{B}_k$ that aggregates statistics over the feature set $\mathcal{F}$: temporal patterns, sensor usage frequencies, and activity durations.

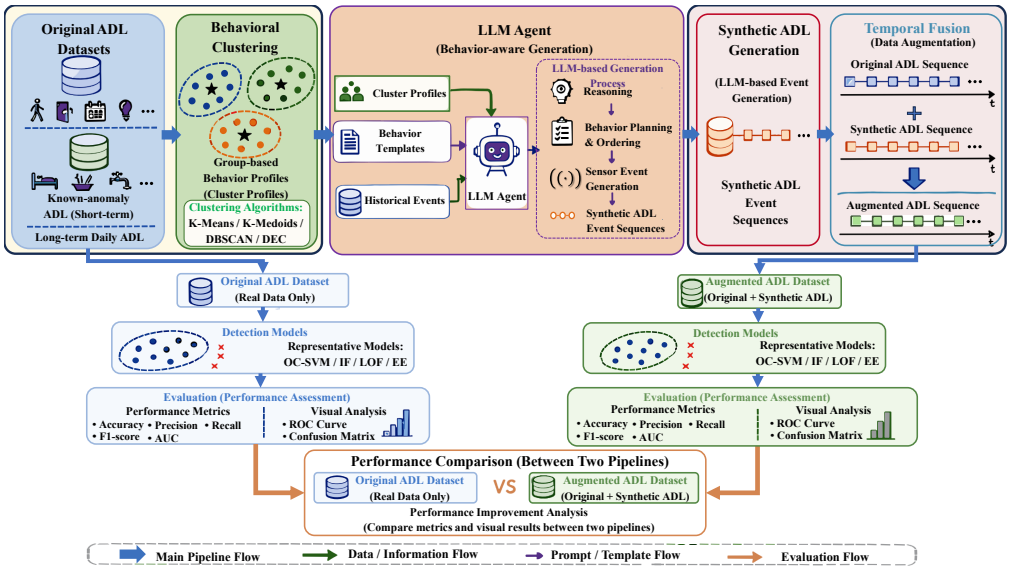

Figure 1: Overview of the proposed Cluster-LLM framework and its downstream evaluation. The framework itself comprises three stages: (1) behavioral clustering partitions the training data $\mathcal{D}_{\text{train}}$ into homogeneous subgroups; (2) LLM-based reasoning generates cluster-conditioned synthetic ADL sequences; and (3) data merging fuses real and synthetic data into the augmented dataset $\mathcal{D}_{\text{aug}}$. The augmented set then feeds (4) one-class anomaly detection (OCSVM, Isolation Forest, LOF, Elliptic Envelope) and (5) multi-metric evaluation comparing original and augmented settings

These profiles serve as the conditioning context for the next stage. By summarizing each subgroup separately rather than the whole dataset, the framework preserves sub-population diversity that a global summary would wash out.

4.2 LLM-Based Synthetic Generation

Building upon the behavioral information extracted in the previous phase, the LLM-based generation module translates cluster-specific statistical signatures into realistic synthetic sensor sequences. We use Qwen-plus Bai et al. (2023) as the LLM backbone, called through the DashScope API with dataset-specific sampling temperatures (0.4 for Kyoto7, 0.7 for HH111) and otherwise default decoding settings; the full API configuration, retry, and parse-failure handling are in Appendix A.4. With the clustering partition $\mathcal{G} = \{G_1, \ldots, G_K\}$ and per-cluster behavioral profiles $\{\mathcal{B}_k\}$ in hand, the next stage turns these profiles into structured synthetic data. The sensor topology $\mathbf{S} = \{s_1, \ldots, s_M\}$ provides spatial context, while two hyperparameters control the generation scale: the volume G (synthetic samples per cluster per activity for Kyoto7) or D (synthetic days for HH111), and the anomaly ratio $r \in [0, 1]$. For each cluster G_k, a structured prompt $\mathcal{P}_k$ assembles four components: a cluster profile $\mathcal{B}_k^{\text{stat}}$ (the statistical summary of G_k over $\mathcal{F}$, directly derived from $\mathcal{B}_k$), the sensor topology $\mathcal{T}$, including locations, message types $\mathcal{M}$, and spatial relationships among functional areas, the activity specification $\mathcal{A}$ (target ADL types, duration ranges $[t_{\min}, t_{\max}]$, and typical event sequences), and the anomaly definition $\mathcal{N}$ (mechanisms such as temporal shifts, duration deviations, sensor omissions, and sequence reorderings). These are concatenated into a single prompt:

$$\mathcal{P}_k = \text{Concat}\big(\mathcal{B}_k^{\text{stat}}, \mathcal{T}, \mathcal{A}, \mathcal{N}\big), \tag{5}$$

and the LLM generates synthetic samples conditioned on $\mathcal{P}_k$: $\tilde{x} \sim g_\theta(\cdot \mid \mathcal{P}_k)$, where g_θ is the generation model with parameters θ. The prompt enforces physical plausibility, strictly increasing timestamps, valid sensor identifiers from $\mathbf{S}$, spatially connected activations, and structured output for automated parsing. At each generation step, a binary flag $a_d \sim \text{Bernoulli}(r)$ decides whether to embed an anomaly scenario $\sigma \in \Sigma$, e.g., bathroom fall, nocturnal wandering, into $\mathcal{P}_k$. The LLM outputs a daily sensor event log $\mathcal{L}_d = \{e_1, e_2, \ldots, e_N\}$ with $N \approx 100$ records, each a tuple $e = (t, s, m)$ satisfying chronological ordering and sensor-state consistency. The complete

synthetic dataset is $\mathcal{D}_{\text{syn}} = \bigcup_{d=1}^{D} \mathcal{L}_d$, where $D \in \{20, 50, 100\}$ for HH111 or the per-cluster volume $G \in \{50, 100, 150, 200\}$ for Kyoto7. Each synthetic log is then mapped to $\tilde{x}$ via the same extractor $\phi(\cdot)$ used for real data, ensuring the synthetic and real samples share a common representation space.

4.3 Anomaly Detection and Evaluation

With $\mathcal{D}_{\text{syn}}$ generated, we merge it with $\mathcal{D}_{\text{train}}$ by simple union: $\mathcal{D}_{\text{aug}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{syn}}$ to train the anomaly detector, where synthetic sequences are integrated into the real ones, and the test set remains strictly composed of genuine data to prevent leakage from the LLM-generated distribution.

In ADL anomaly detection, routine behaviors dominate the observed distribution, whereas anomalous events remain extremely scarce and heterogeneous. Our Cluster-LLM solves this challenge in two ways: it synthesizes plausible normal trajectories to broaden in-distribution support, while generating boundary-adjacent anomalies to sharpen decision thresholds. Consequently, we train four one-class classifiers on $\mathcal{D}_{\text{aug}}$, each learning the decision function $f : \mathbb{R}^d \to \{0, 1\}$ from Section 3.1. The four span complementary geometric assumptions. OCSVM Schölkopf et al. (2001); Tax & Duin (2004) fits a tight boundary around the normal region in a high-dimensional space; it overfits when training data is sparse—the gap our augmentation fills. iForest Liu et al. (2008; 2012) isolates anomalies via random partitioning trees, tolerating high-dimensional, noisy features. LOF Breunig et al. (2000) flags points whose local density falls well below their neighbors', catching deviations within a subgroup rather than across the whole population. EE Rousseeuw (1985); Hardin & Rocke (2005) fits a robust covariance ellipse under a Gaussian assumption, offering a parametric counterpoint to the other three. Running all four separates gains from augmentation versus detector-specific quirks: OCSVM excels on Kyoto7's compact features, while iForest and LOF suit HH111's longer temporal sequences.

The training procedure differs by setting. For Kyoto7, stratified cross-validation Kohavi (1995) preserves the anomaly ratio across folds; $\mathcal{D}_{\text{aug}}$ includes both synthetic normal sequences and injected anomalies (governed by r), exposing the classifier to patterns real data alone cannot cover. For HH111, we adopt a temporal split, where earlier historical records form the training distribution, while subsequent periods serve as the evaluation set. This setup emulates real-world deployment by requiring models to detect future behavioral anomalies based solely on historical baselines.

Given the severe class imbalance in anomaly detection, standard accuracy fails to reflect true detection capability. We evaluate all augmented detectors using a comprehensive suite of seven metrics, combining standard performance measures with imbalance-robust criteria. Complete metric definitions, formal formulations, and the comparative protocol are detailed in Section 5.

5 Experiments

In this section, we conduct extensive evaluations to benchmark the proposed Cluster-LLM framework across two complementary smart home environments: *Kyoto7* and *HH111*. To assess the framework's versatility, Cluster-LLM is instantiated using four distinct behavioral clustering backbones to construct subgroup profiles. Downstream performance is evaluated using four representative one-class classifiers trained on the augmented datasets and compared against non-augmented baselines across seven comprehensive performance metrics. Our experimental design is governed by three central Research Questions (RQs): **RQ1:** Does cluster-conditioned synthetic augmentation consistently enhance downstream anomaly detection performance compared to non-augmented baselines across both known-anomaly and unsupervised settings? (see Figure C1, Figure C2, Table C1, Table C2, Table D1, Table D2, Table D3) **RQ2:** How do different behavioral clustering paradigms impact prompt conditioning and downstream classifier generalization?(see Figure D1, Figure C1, Figure C2, Figure D8, Figure D2, Figure D4, Figure D6, Figure D3, Figure D5, Figure D7, Table D6, Table D5, Table C2, Table B1, Table B2) **RQ3:** How does the volume of generated synthetic data affect model generalization, and what is the optimal real-to-synthetic ratio to prevent generation-induced distribution shift?(see Figure D1, Figure C2, Figure D8, Table C1, Table D6, Table D5, Table D3)

5.1 Experimental Setup

Datasets. We evaluate on two smart home datasets from the CASAS repository Cook et al. (2009); Cook (2012); Cook et al. (2013), representing two distinct anomaly detection paradigms.

Kyoto7 provides the known-anomaly setting. It comes from a single-resident CASAS testbed with 30 ambient sensors, where dozens of participants each performed the same five activities (phone call, hand washing, cooking, eating, cleaning); one scripted error is attached to each activity, giving five labeled anomaly types. Raw sensor events are tuples $e = (t, s, m)$ of timestamp, sensor ID, and status message. We evaluate generative scaling at volumes $G \in \{50, 100, 150, 200\}$ synthetic instances per cluster per activity.

Another dataset is **HH111**, comprising 61 consecutive days of continuous sensor streams (motion, door, temperature, light) from a single-resident home. The release carries no per-day anomaly annotations, so we derive day-level labels from the stream through a transparent rule set described below; this follows the established practice of labeling HH111 from the data itself rather than from shipped ground truth Yahaya et al. (2021b). Synthetic traces are generated across temporal windows $D \in \{20, 50, 100\}$ days, in which a controlled fraction $r = 0.3$ of the generated sequences is anomalous to match the anomaly share seen in the real routine. HH111 ships without labels, so we derive day-level anomaly labels from the stream itself, and fix one evaluation split shared by all variants: 10 real days enter training, the remaining 51 real days form the test set, and generated days go to training only. Appendix A.1 and A.2 gives the full labeling rules and split procedure.

Evaluation Metrics. Smart-home anomaly detection is dominated by normal days, so raw Accuracy is misleading: a model that always predicts "normal" can score high while missing every true anomaly. We therefore report seven complementary metrics. Four are standard classification measures—Accuracy, Precision, Recall, and F1-score—and three target extreme class imbalance: **Balanced Accuracy (Bal Acc)** Brodersen et al. (2010), which averages per-class recall to remove majority-class bias; **Matthews Correlation Coefficient (MCC)** Chicco & Jurman (2020), which weighs all four confusion-matrix cells and stays informative under heavy skew; and **Area Under the ROC Curve (AUC)** Fawcett (2006), which scores threshold-free ranking quality. No single number tells the whole story; together they show whether augmentation genuinely improves detection or merely shuffles errors. For every combination of clustering method, generation volume, and detection model, we compare the original baseline ($\mathcal{D}_{\text{train}}$ only) against the augmented setting ($\mathcal{D}_{\text{aug}}$).

Implementation Details. The proposed Cluster-LLM framework and all baseline architectures are implemented in Python 3.10. Anomaly detection algorithms, feature preprocessing, and baseline models are executed using `scikit-learn` Pedregosa et al. (2011), while Deep Embedded Clustering (DEC) is constructed using TensorFlow/Keras. For generative augmentation we call Qwen-plus through the DashScope API; its full configuration follows Appendix A.4. The number of clusters K for centroid-based (K-Means, K-Medoids) and deep latent (DEC) models is selected with the Silhouette Coefficient. DBSCAN parameters (ϵ, MinPts) are grid-searched until the partition yields the same number of routines used by the other backbones; we state this explicitly because it partially offsets DBSCAN's appeal of not needing a preset cluster count. The complete hyperparameter settings for all detection models are listed in Table A1.

5.2 Results Analysis

Effectiveness of Cluster-LLM (RQ1). We conduct extensive evaluations across two smart home environments: *Kyoto7* and *HH111*. Under the Kyoto7 setting, the results in Table 1 reveals distinct performance dynamics conditioned on activity type and behavioral variance. For flexible, high-variance routines such as phone calls, Cluster-LLM delivers substantial detection gains across downstream architectures. Specifically, the EE model achieves a peak F1-score of 0.667 at $G = 100$ under K-means, DBSCAN, and DEC clustering paradigms, yielding a 50.2% relative enhancement over the same detector trained without augmentation (0.444, the "Original" row of Table 1). Similarly, OCSVM exhibits consistent performance gains, improving from 0.500 to 0.571 under K-means and K-Medoids. These improvements are strongly corroborated by imbalance-robust metrics reported in Table B1, where the Balanced Accuracy and AUC both increase from 0.500 to 0.625 across K-means, DBSCAN, and DEC configurations, while the MCC leaps from 0.000 to 0.250.

Conversely, highly structured routines with strict sequential constraints, such as hand-washing, exhibit more modest responsiveness to synthetic augmentation. As detailed in Tables B2 and C1, EE's peak F1-score rises modestly from 0.476 to 0.516, with Balanced Accuracy and AUC increasing from 0.444 to 0.556, while LOF suffers from zero recall across all augmented conditions. HH111 tells a more conditional story, as shown in Table 2. At $D = 20$ days, accuracy climbs across all

Table 1: Kyoto7 phone-call detection. Each cell reports the test-set result of a one-class detector trained on real activity instances plus G synthetic instances per cluster, with activity instances from held-out tasks as evaluation units. "Original" is the same detector trained on real data only; all relative gains in the text use this row as baseline.

Clustering	G	OCSVM			iForest			LOF			EE		
		Acc	Rec	F1	Acc	Rec	F1	Acc	Rec	F1	Acc	Rec	F1
K-means	50	0.500	0.667	0.500	0.250	0.667	0.400	0.375	0.667	0.444	0.375	0.667	0.444
	100	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.333	0.400	**0.750**	0.667	0.667
	150	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.667	**0.571**	0.375	0.667	0.444
	200	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.667	**0.571**	0.375	0.667	0.444
K-Medoids	50	0.500	0.667	0.500	**0.375**	0.667	**0.444**	0.375	0.667	0.444	0.375	0.667	0.444
	100	**0.625**	0.667	**0.571**	**0.375**	0.667	**0.444**	0.375	0.333	0.286	**0.500**	0.333	0.333
	150	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.667	**0.571**	0.375	0.667	0.444
	200	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.667	**0.571**	0.375	0.667	0.444
DBSCAN	50	0.375	0.333	0.286	**0.500**	0.667	**0.500**	**0.625**	0.333	0.400	**0.750**	0.667	0.667
	100	0.500	0.000	0.000	0.250	0.667	0.400	**0.750**	0.667	**0.667**	**0.750**	0.667	0.667
	150	**0.625**	0.333	0.400	**0.500**	0.667	**0.500**	**0.625**	0.333	0.400	**0.750**	0.667	0.667
	200	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.667	**0.571**	0.375	0.667	0.444
DEC	50	0.375	0.667	0.444	**0.500**	0.667	**0.500**	0.375	0.667	0.444	0.250	0.667	0.400
	100	0.500	0.333	0.333	**0.625**	0.667	**0.571**	**0.625**	0.333	0.400	**0.750**	0.667	0.667
	150	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.667	**0.571**	0.375	0.667	0.444
	200	**0.625**	0.667	**0.571**	**0.500**	0.667	**0.500**	**0.625**	0.667	**0.571**	0.375	0.667	0.444
Original	–	0.500	0.667	0.500	0.250	0.667	0.400	0.375	0.667	0.444	0.375	0.667	0.444

backbones: OCSVM goes from 0.721 to 0.759 under DEC, but F1 falls below the original 0.267 for every clustering method (the added volume is too thin to reshape the decision boundary, and precision drops as false alarms accumulate). F1 first clears the baseline only at $D = 50$ days (K-Means, 0.274). The pattern across both datasets is that augmentation pays off where behavioral variance is high and within a volume window that depends on the dataset itself.

Impact of Clustering Approach (RQ2). We then evaluate how distinct partitioning paradigms, that are K-Means, K-Medoids, DBSCAN, and DEC, influence LLM prompt conditioning and downstream classifier generalization. On Kyoto7, as shown in Table C2, for high-variance activities like phone calls, K-Means, DBSCAN, and DEC achieve an identical peak F1-score of 0.667 ($+50.2\%$ relative improvement over baseline), whereas K-Medoids exhibits a more constrained gain. For highly deterministic routines like hand-washing, all four paradigms converge exactly to $F1 = 0.516$ ($+8.4\%$). This invariant convergence confirms that when underlying behavioral variation is tightly bounded, downstream generalization becomes largely insensitive to the choice of clustering backbone.

On the unsupervised HH111 benchmark, performance rankings diverge depending on temporal volumes and metric priorities, as shown in Table D4 and Table D5. At $D = 50$ days, K-Means establishes the top F1-score, whereas K-Medoids yields a lower F1-score (0.179) but secures the highest overall classification accuracy. Density-based partitioning (DBSCAN) demonstrates a specialized inductive synergy with local spatial estimators, elevating LOF's F1-score from 0.097 to 0.221 and effectively restoring discriminative capability to an otherwise failing detector. Additionally, deep latent clustering (DEC) delivers superior accuracy at smaller generation volumes but experiences performance decay as synthetic volume expands. These empirical trade-offs demonstrate that optimal backbone selection is governed by the alignment between the topological assumptions of the clustering algorithm and the inductive biases of downstream anomaly detectors.

Effect of Synthetic Data Volume (RQ3). We investigate the performance across varying augmentation volumes to delineate the non-monotonic trade-offs between synthetic generation quantity and classifier generalization. On Kyoto7 (Figure C2), downstream detection metrics exhibit clear asymptotic plateaus as the sample-level volume G increases, indicating that once cluster-conditioned synthetic traces cover the underlying behavioral sub-manifold, additional generation produces diminishing marginal returns and redundant computational overhead.

Table 2: HH111 anomaly detection results with 20-day augmentation ($D = 20$, $r = 0.3$). Each row shows a detection model and metric, with columns for the original baseline and four clustering-guided augmentation methods.

Model	Metric	Orig	K-means	K-Medoids	DBSCAN	DEC
OCSVM	Acc	0.721	**0.748**	**0.736**	**0.736**	**0.759**
	Prec	0.661	0.556	0.588	0.604	0.535
	Rec	0.167	0.099	0.110	0.112	0.087
	F1	0.267	0.169	0.186	0.189	0.149
iForest	Acc	0.705	**0.737**	**0.724**	**0.728**	**0.748**
	Prec	0.577	0.452	0.467	0.524	0.379
	Rec	0.118	0.115	0.082	0.101	0.051
	F1	0.196	0.183	0.140	0.169	0.090
LOF	Acc	0.670	**0.709**	**0.702**	**0.701**	**0.733**
	Prec	0.289	0.171	0.241	0.229	0.235
	Rec	0.058	0.034	0.042	0.038	0.042
	F1	0.097	0.057	0.072	0.065	0.072
EE	Acc	0.671	**0.732**	**0.723**	**0.714**	**0.721**
	Prec	0.254	**0.392**	**0.469**	**0.376**	0.153
	Rec	0.042	**0.073**	**0.099**	**0.067**	0.032
	F1	0.072	**0.123**	**0.164**	**0.112**	0.053

On the HH111 benchmark, this non-monotonic trajectory manifests as a sharp inverted U-shape. At a temporal volume of $D = 20$ days, insufficient representation density prevents models from surpassing the non-augmented baseline F1-score of 0.267. Expanding volume to $D = 50$ days establishes optimal manifold coverage, with K-Means achieving a peak F1 of 0.274 and overall accuracy reaching 0.811 (Table D5). However, pushing volume to $D = 100$ days triggers severe performance degradation across all paradigms. This volume-dependent threshold is vividly illustrated in cross-volume comparisons (Figure D8) and learning curve analyses (Figure D3, Figure D5, Figure D7), where model F1-scores crest at an intermediate training fraction before continuously eroding under larger sample sizes. The mechanistic driver of this divergence lies in the structural conditioning of the datasets rather than sample size alone. In Kyoto7, discrete, activity-conditioned prompt constraints bound the LLM within specific event spaces, allowing synthetic samples to safely populate sparse boundary regions without corrupting class priors. In contrast, HH111 relies on unconstrained continuous stream generation in a fully unsupervised environment. At $D = 100$ days, synthetic samples constitute over 83% of the training set, causing downstream estimators to fit LLM generation artifacts rather than authentic human routines. Consequently, optimal generative augmentation requires precisely calibrating synthetic volume against underlying manifold complexity, as exceeding the saturation threshold induces severe generation-induced distribution shift.

6 Conclusion

This paper presents Cluster-LLM, a novel data augmentation framework that integrates behavioral topology clustering with LLM-conditioned sequence reasoning for smart home anomaly detection. Cluster-LLM effectively mitigates synthetic data homogeneity and alleviates extreme class imbalance. We identify that behavioral diversity dictates gain, where high-variance activities benefit far more from generative manifold expansion than rigid, deterministic routines. Exceeding optimal augmentation thresholds introduces distribution shift, particularly in unsupervised single-resident settings. These insights provide guidelines for deploying generative models in real-world applications. In the future, we will extend this framework toward multi-resident co-habitation topology modeling, domain-specific behavioral latent embeddings, active learning mechanisms for iterative prompt alignment, and multi-label temporal anomaly detection.

AI use statement

We used large language models in two ways. First, as a core methodological component, we employ the Qwen-plus API to generate the synthetic smart-home sensor-event sequences that are the subject

of this paper (Section 3.3); this use is fully disclosed in the Methodology and is not an assistive writing tool. Second, we used generative-AI assistants for limited language proofreading of the manuscript; all suggestions were reviewed and verified by the authors, and we take full responsibility for the final text, claims, and results. The LLM-generated synthetic data was validated by checking feature-space consistency against real sequences and by the downstream detection improvements reported in the experiments; we did not use generative AI to fabricate results, citations, or data.

ETHICS STATEMENT

The datasets used (CASAS Kyoto7 and HH111) are publicly released smart-home sensor benchmarks collected for ambient-assisted-living research. They contain no newly recruited human subjects and no personally identifiable information beyond de-identified sensor event logs, and they required no IRB-approved human-subjects procedures. All anomalous scenarios (e.g., falls, nocturnal wandering) are either scripted in the benchmark or synthetically generated, not derived from real individuals' medical records. Smart-home sensor data is nonetheless privacy-sensitive; we use only the public benchmark releases and do not collect or redistribute additional personal data. We acknowledge a potential dual-use risk that synthetic sensor traces could be used to craft deceptive sequences, but our pipeline is constrained to the two benchmark datasets and is intended to improve safety monitoring for older adults.

REPRODUCIBILITY STATEMENT

We release the full implementation and data-processing pipeline to support reproducibility. The code follows a five-stage sequence—data preprocessing, clustering, LLM-based generation, data connection, and training/detection—organized under `code/` with per-dataset (Kyoto7/HH111) and per-stage subdirectories; the included README specifies the execution order and the directory tree. All scripts locate data and dependencies through relative paths, so the directory structure must be preserved. Dependencies are Python 3.9+ with pandas, numpy, scikit-learn, matplotlib, seaborn, and openai; DEC clustering additionally requires TensorFlow. Synthetic data is produced through the Qwen-plus API using a key supplied in `code/api_key.txt` (Stage 3 is billed per API call). Hyperparameter settings for all detection models are listed in Table A1; the complete notation table is Table A2. Full per-volume results for Kyoto7 and HH111 appear in Appendix B–D (Kyoto7: Tables B1–B2, Figures C1–C2, and Tables C1–C2; HH111: Tables D1–D6 and Figures D1–D8), including ROC and learning curves. The exact experimental matrix (clustering methods $\times$ generation volumes) is summarized in the code README. Both datasets are public CASAS benchmarks, so no additional data is required to reproduce the experiments.

REFERENCES

E. Achirei and L. Alboaie. Development and evaluation of a passive monitoring system for the elderly using the matter protocol. *Procedia Computer Science*, 246:2274–2283, 2024.

Q. Ai, P. Wang, L. He, L. Wen, L. Pan, and Z. Xu. Generative oversampling for imbalanced data via majority-guided vae. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 3315–3330, 2023.

C. V. Anikwe, H. F. Nweke, Y. W. Teh, et al. Mobile and wearable sensors for data-driven health monitoring system: State-of-the-art and future prospect. *Expert Systems with Applications*, 202:117362, 2022.

J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, et al. Qwen technical report, 2023.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000.

K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pp. 3121–3124, 2010.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

S. Chernbumroong, S. Cang, A. Atkins, and H. Yu. Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, 40(5):1662–1674, 2013.

G. Chicco and M. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, 2020.

D. J. Cook. Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems*, 27(1): 32–38, 2012.

D. J. Cook, M. Schmitter-Edgecombe, A. Crandall, et al. Collecting and disseminating smart home sensor data in the casas project. In *Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets*, pp. 1–8, 2009.

D. J. Cook, A. Crandall, B. Thomas, and N. Krishnan. Casas: A smart home in a box. *IEEE Computer*, 46(7): 62–69, 2013.

S. G. Dhekane and T. Plötz. Transfer learning in sensor-based human activity recognition: A survey. *ACM Computing Surveys*, 57(8):1–39, 2025.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

H. Fu, H. Chen, S. Lin, and G. Xing. Shade-ad: An llm-based framework for synthesizing activity data of alzheimer's patients. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 290–296, 2025.

J. Hardin and D. M. Rocke. The distribution of robust distances. *Journal of Computational and Graphical Statistics*, 14(4):928–946, 2005.

H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1322–1328, 2008.

V. R. Jakkula and D. J. Cook. Detecting anomalous sensor events in smart home data for enhancing the living experience. In *Proceedings of the AAAI Workshop on Plan, Activity, and Intent Recognition (PAIR)*, 2011.

L. Kaufman and P. J. Rousseeuw. Clustering by means of medoids. In Y. Dodge (ed.), *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pp. 405–416. North-Holland, 1987.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1137–1145, 1995.

Z. Leng, M. Thukral, Y. Liu, H. Rajasekhar, S. K. Hiremath, J. He, and T. Plötz. Agentsense: Virtual sensor data generation using llm agents in simulated home environments, 2025.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 413–422, 2008.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 6(1):1–39, 2012.

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

A. Nawaz et al. Fall detection system using tabular gan for data augmentation with integration of isolation forest model. *Applied Soft Computing*, 185:113931, 2025. doi: 10.1016/j.asoc.2025.113931.

R. Ohlan and A. Ohlan. A comprehensive bibliometric analysis and visualization of smart home research. *Technological Forecasting and Social Change*, 184:121975, 2022. doi: 10.1016/j.techfore.2022.121975.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

B. Prenkaj, P. Velardi, et al. A self-supervised algorithm to detect signs of social isolation in the elderly from daily activity sequences. *Artificial Intelligence in Medicine*, 135:102454, 2023. doi: 10.1016/j.artmed.2022.102454.

J. M. N. Ramirez and Y. Cardinale. Activities of daily living detection on healthcare: A categorization. In *Proceedings of the 7th International Workshop on Sensor-based Activity Recognition and Artificial Intelligence (iWOAR), Article 6*, pp. 1–10, 2022.

Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, P. S. Yu, and L. He. Deep clustering: A comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):5858–5878, 2025.

D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui. Smartfaber: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment. *Artificial Intelligence in Medicine*, 67: 57–74, 2016.

P. J. Rousseeuw. Multivariate estimation with high breakdown point. In W. Grossmann, G. Pflug, I. Vincze, and W. Wertz (eds.), *Mathematical Statistics and Applications*, volume B, pp. 283–297. Reidel, Dordrecht, 1985.

N. Roy, N. Pathak, and A. Misra. Fine-grained appliance usage and energy monitoring through mobile and power-line sensing. *Pervasive and Mobile Computing*, 30:132–150, 2016. doi: 10.1016/j.pmcj.2016.01.003.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

E. Schubert and P. J. Rousseeuw. Fast and eager k-medoids clustering: $o(k)$ runtime improvement of the pam, clara, and clarans algorithms. *Information Systems*, 101:101804, 2021.

D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

N. C. Tay, T. Connie, T. S. Ong, A. B. J. Teoh, and P. S. Teh. A review of abnormal behavior detection in activities of daily living. *IEEE Access*, 11:5069–5088, 2023.

Y. Tong, R. Chen, and J. Gao. Hidden state conditional random field for abnormal activity recognition in smart homes. *Entropy*, 17(3):1358–1378, 2015.

J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 478–487, 2016.

S. W. Yahaya, A. Lotfi, and M. Mahmud. A consensus novelty detection ensemble approach for anomaly detection in activities of daily living. *Applied Soft Computing*, 83:105613, 2019.
S. W. Yahaya, A. Lotfi, and M. Mahmud. Towards a data-driven adaptive anomaly detection system for human activity. *Pattern Recognition Letters*, 145:200–207, 2021a.
S. W. Yahaya, A. Lotfi, and M. Mahmud. Detecting anomaly and its sources in activities of daily living. *SN Computer Science*, 2(1):14, 2021b.
W. You, H. Jiang, Z. Liu, Z. Xie, T. Liu, J. Lu, and F. Dou. Adlgen: Synthesizing symbolic, event-triggered sensor sequences for smart-home human activity modeling. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 1029–1042, 2026.
F. Zhang, A. P. C. Chan, and D. Li. Developing smart buildings to reduce indoor risks for safety and health of the elderly: A systematic and bibliometric analysis. *Safety Science*, 168:106310, 2023. doi: 10.1016/j.ssci.2023.106310.

A Dataset Protocols, Implementation Details, and Notation

Figure A1 illustrates the smart home ADL monitoring pipeline. Panel (a) shows a typical deployment: ambient sensors—motion detectors, door contacts, pressure mats—are distributed across functional areas including the kitchen, living room, bedroom, bathroom, and entrance. Each sensor fires discrete events when state changes occur. Panel (b) traces the data flow: raw sensor events $e_i = (t_i, s_i, m_i)$ are collected within a temporal window, aggregated into an activity instance $\xi = [e_1, e_2, \ldots, e_L]$, and mapped to a feature vector $x = \phi(\xi)$ via extractor $\phi(\cdot)$. The anomaly detector $f(x)$ then classifies the instance as normal or anomalous. This pipeline underlies all experiments in the main paper.

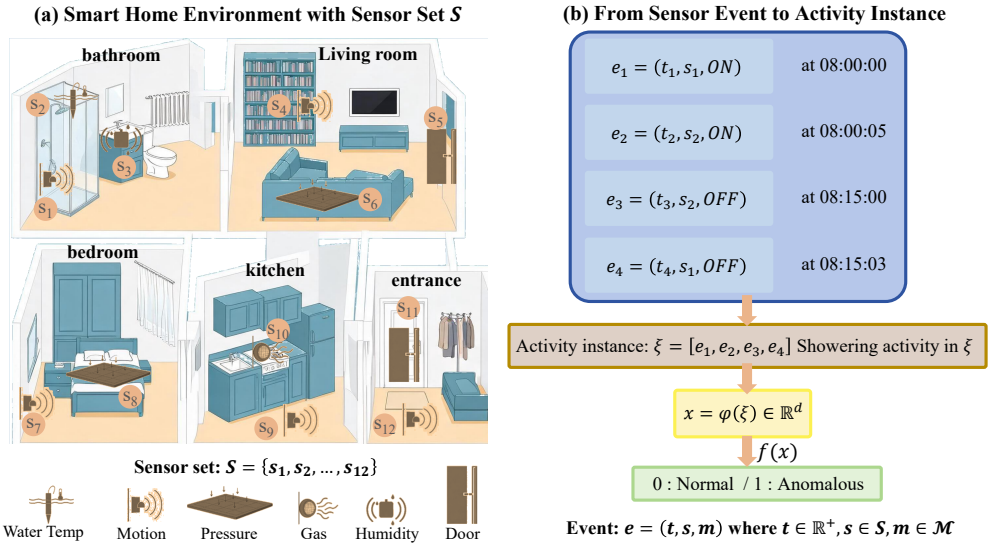

Figure A1: Illustration of the smart-home ADL monitoring scenario. **(a)** smart home environment with sensor set $\mathbf{S} = \{s_1, \ldots, s_{12}\}$ deployed across five functional areas. **(b)** from sensor events to activity instance: discrete sensor events $e_i = (t_i, s_i, m_i)$ are aggregated into an activity instance $\xi = [e_1, e_2, e_3, e_4]$, which is mapped to a feature vector $x = \phi(\xi) \in \mathbb{R}^d$ via a feature extractor $\phi(\cdot)$, and classified by the decision function $f(x)$ as normal or anomalous.

A.1 HH111 Labeling Protocol

CASAS HH111 comes without anomaly labels. Prior work labels it by filtering to a target activity and injecting anomalies, then marking the changed features as the source Yahaya et al. (2021b). We do the same—labeling from the data itself rather than trusting pre-supplied ground truth—but at the day level, which fits our one-class detectors. For each day we compute the event count, the night-time volume, and the longest daytime silence, then flag it as anomalous when it clearly breaks the resident's usual routine: an unusual night-time spike, a daytime gap past a threshold, or a daily total more than a few standard deviations off the mean. Our two rules watch the sleep pattern, the same axis

Yahaya et al. call the sleeping activity and label *Less sleeping* or *Interrupted sleep*; a flagged day is therefore a sleep anomaly, not a random event burst. We attach the binary label y_i only to real days and keep it out of training, as the unsupervised setup in Section 3.1 requires. One granularity difference: Yahaya et al. label at the feature level and name the exact source (start time, duration, interruption). Our detectors take a whole day as one instance, so they need a single normal/abnormal verdict; we collapse to a day-level label. The two conventions mean the same thing, just at different resolutions.

A.2 HH111 Evaluation Protocol

The 61 real days are combined with the $D \in \{20, 50, 100\}$ generated days into one pool of $61 + D$ days. A straight 50/50 split would put almost no real days in training (only 5 when $D = 50$, none when $D = 100$), so we split the real days directly: 10 go to training and the other 51 form a fixed test set used by every variant. The D generated days join training only. Training then has $10 + D$ days and always at least 10 real ones (30, 60, 110 for $D = 20, 50, 100$), while the test set stays at 51 real days whatever D is. All detectors are scored on this same held-out real stream.

A.3 Detailed Hyperparameter Settings

Table A1 lists the hyperparameter settings for each detection model.

Table A1: Hyperparameter settings for anomaly detection models.

Model	Hyperparameter	Value
OCSVM	Kernel	RBF
	ν	0.5
	γ	scale
iForest	n_estimators	100
	max_samples	auto
	contamination	auto
LOF	n_neighbors	20
	algorithm	auto
	contamination	auto
EE	support_fraction	None
	contamination	0.1

A.4 LLM Backbone and Generation Configuration

All synthetic sequences come from Qwen-plus Bai et al. (2023), called through the DashScope compatible-mode API at `https://dashscope.aliyuncs.com/compatible-mode/v1` (model `qwen-plus`; calls made from mainland China, June–August 2025). We tune the sampling temperature per dataset: 0.4 on Kyoto7, whose scripted activities call for staying close to the cluster statistics, and 0.7 on HH111, where looser sampling adds day-level variety. All other decoding settings use the API defaults. Failed or malformed calls are retried up to three times. Any record that still will not parse as JSON or is missing fields is dropped; we log the per-run parse-failure rate with the released code so the exact generated count stays reproducible.

A.5 Notation

The paper introduces a large set of mathematical symbols across the problem formulation and methodology—sensors, features, datasets, clustering, prompts, and generation parameters all carry their own notation. Table A2 collects every symbol in one place so the reader can look up any notation without flipping back and forth between sections. Descriptions are kept brief; the full formal definitions remain in the main text.

The notation in Table A2 spans the full pipeline—from raw sensor events through clustering and prompt construction to the final generation outputs. Readers familiar with the smart home domain

Table A2: Summary of notation used in this paper. Symbols appear in the order they are introduced in sections 3 and 4.

Symbol	Description	Symbol	Description
$\mathbf{S} = \{s_1, \ldots, s_M\}$	Set of M ambient sensors deployed in the smart home	ϵ	Neighborhood radius defining local density (DBSCAN)
s	Individual sensor identifier within $\mathbf{S}$	MinPts	Minimum points required to form a dense region
M	Total number of sensors	$\mathcal{O}_{\text{noise}}$	Noise points in low-density regions
$e = (t, s, m)$	Sensor event: timestamp, sensor ID, and status message	$f_\phi : \mathbb{R}^d \to \mathcal{Z}$	Non-linear encoder mapping features to latent space (DEC)
$t \in \mathbb{R}^+$	Continuous timestamp of a sensor event	$\mathcal{Z} \subset \mathbb{R}^{d'}$	Low-dimensional latent representation space
$m \in \mathcal{M}$	Sensor reading, e.g., `ON`/`OFF`	d'	Dimensionality of the latent space
$\mathcal{M}$	Set of all admissible status messages	$\mathbf{z}_i = f_\phi(\mathbf{v}_i)$	Latent representation of instance i
$\xi = [e_1, \ldots, e_L]$	Temporal sequence of L sensor events (one activity)	q_{ik}	Soft assignment of instance i to cluster k (Student's t)
L	Number of events in one activity instance	α	Degrees of freedom of the Student's t-distribution
$\phi(\cdot)$	Mapping from activity sequence to feature vector	$P = [p_{ik}]$	Auxiliary target distribution for KL-divergence
$x = \phi(\xi) \in \mathbb{R}^d$	d-dim feature vector (spatiotemporal and transition stats)	$Q = [q_{ik}]$	Soft assignment matrix
$\mathbf{v}_i$	Feature vector of instance i (clustering notation)	$\mathcal{L}_{\text{DEC}}$	KL-divergence loss between P and Q
d	Dimensionality of the feature vector	$\mathcal{B}_k$	Aggregated stats (temporal, frequency, duration) for G_k
$\mathcal{D}_{\text{real}}$	Real-world dataset of N labeled instances	$\mathcal{B}_k^{\text{stat}}$	Statistical summary of G_k, injected into the prompt
N	Number of real instances	$\mathcal{F}$	Feature set used for behavioral profiling
$y_i \in \{0, 1\}$	Anomaly label (0 = normal, 1 = anomalous)	$\mathcal{T}$	Sensor topology: locations, types, spatial relations
$f : \mathbb{R}^d \to \{0, 1\}$	Decision function of the anomaly detector	$\mathcal{A}$	Activity spec: ADL types, durations, typical sequences
$\mathcal{D}_{\text{syn}}$	LLM-generated synthetic feature vectors	$\mathcal{N}$	Anomaly definition: temporal shifts, omissions, etc.
N_{syn}	Number of synthetic instances	$\mathcal{P}_k$	Structured prompt for cluster G_k
$\tilde{x}$	A single synthetic feature vector	$g_\theta(\cdot \mid \mathcal{P}_k)$	LLM generation model, conditioned on $\mathcal{P}_k$
$\mathcal{D}_{\text{aug}}$	Merged set: $\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{syn}}$	θ	Parameters of the LLM generation model
K	Number of clusters	G	Samples per cluster per activity (Kyoto7: $\{50, 100, 150, 200\}$)
$\mathcal{G} = \{G_1, \ldots, G_K\}$	Partition of $\mathcal{D}_{\text{real}}$ into K subgroups	D	Synthetic days generated (HH111: $\{20, 50, 100\}$)
G_k	The k-th cluster subset	$r \in [0, 1]$	Anomaly injection ratio
$\boldsymbol{\mu}_k$	Mean centroid of cluster G_k (K-means)	$a_d \sim \text{Bernoulli}(r)$	Binary flag: whether day d gets an anomaly
$\mathbf{m}_k$	Exemplar medoid of G_k (K-Medoids)	$\sigma \in \Sigma$	Anomaly scenario (e.g., fall, wandering)
$d(\cdot, \cdot)$	Pairwise distance metric	Σ	Set of all anomaly scenarios
$N_\epsilon(\mathbf{v})$	ϵ-neighborhood: points within distance ϵ	$\mathcal{L}_d$	Daily sensor event log from the LLM (≈ 100 records)
		$[t_{\min}, t_{\max}]$	Permissible duration range for a target activity

will recognize the sensor-level notation ($\mathbf{S}$, e, ξ) as fairly standard. The clustering block (K, $\boldsymbol{\mu}_k$, $\mathbf{m}_k$, N_ϵ, q_{ik}, $\mathcal{L}_{\text{DEC}}$) covers the four paradigms we benchmark, so the same symbol family appears across K-means, K-Medoids, DBSCAN, and DEC. The prompt and generation block ($\mathcal{P}_k$, g_θ, G, D, r) is specific to our framework and worth keeping handy when reading the Methodology section. If a symbol's meaning seems ambiguous in context, the precise definition is always in the main text.

B Additional Evaluation Metrics for Kyoto7

Tables B1 and B2 report the complete evaluation results including Balanced Accuracy, MCC, and AUC for the Kyoto7 dataset. These metrics complement the Accuracy, Recall, and F1-score reported in the main paper. Under severe class imbalance—where anomalies constitute a small fraction of the data—Accuracy alone can mask poor detection. Balanced Accuracy corrects this bias by averaging per-class recall. MCC captures the correlation between predicted and true labels across all four confusion-matrix cells, and AUC measures ranking quality independent of any single threshold. Together they paint a fuller picture of how augmentation shifts the detection landscape.

Table B1: Kyoto7 phone call: balanced accuracy, MCC, and AUC for the best performing configuration per clustering method. The baseline row reports scores without any synthetic augmentation.

Method	**Bal Acc**	**MCC**	**AUC**
Original	0.500	0.000	0.500
K-means	0.625	0.250	0.625
K-Medoids	0.563	0.143	0.563
DBSCAN	0.625	0.250	0.625
DEC	0.625	0.250	0.625

Table B2: Kyoto7 hand-washing: balanced accuracy, MCC, and AUC for the best performing configuration per clustering method. Identical scores across all four methods reflect the tight behavioral structure of this activity.

Method	**Bal Acc**	**MCC**	**AUC**
Original	0.444	0.023	0.444
K-means	0.556	0.053	0.556
K-Medoids	0.556	0.053	0.556
DBSCAN	0.556	0.053	0.556
DEC	0.556	0.053	0.556

Tables B1 and B2 show that clustering-guided augmentation lifts every imbalanced-robust metric over the original baseline. For phone call, K-means, DBSCAN, and DEC push Balanced Accuracy from 0.500 to 0.625 and AUC from 0.500 to 0.625. The MCC jump from 0.000 to 0.250 means the augmented models catch true positives without flooding the user with false alarms. K-Medoids lands in between, reaching Bal Acc 0.563 and MCC 0.143—still better than the baseline, but the medoid constraint apparently limits how much behavioral variety the LLM can inject. For hand-washing, all four methods produce identical gains (Bal Acc: 0.444 $\rightarrow$ 0.556, MCC: 0.023 $\rightarrow$ 0.053). The uniformity here is telling: when an activity has a rigid event sequence, the choice of clustering backbone matters less than for flexible routines. The smaller MCC values (0.053 vs. 0.250) also confirm that hand-washing detection remains a harder problem even after augmentation.

C Kyoto7 Additional Results

C.1 F1-Score Comparison Across Clustering Methods

Figure C1 visualizes the best F1 scores achieved by each clustering method on the Kyoto7 dataset. The dashed red line marks the original baseline without augmentation, so any bar rising above it signals a genuine gain.

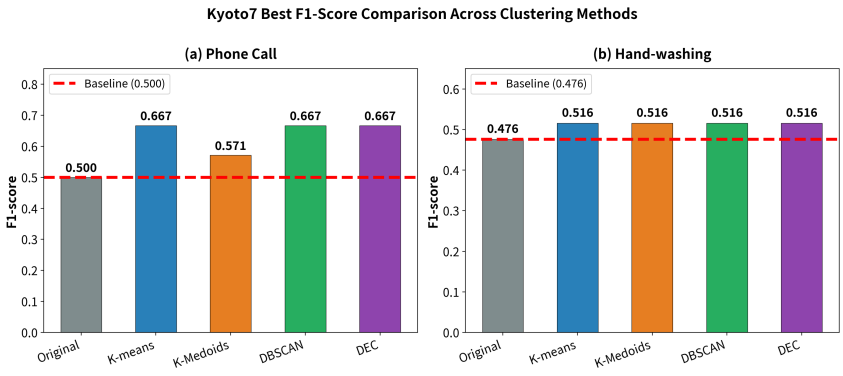

Figure C1: Kyoto7 best F1-score comparison across clustering methods for (a) phone call and (b) hand-washing anomaly detection. The dashed red line indicates the original baseline without augmentation. Bars above the line represent improvements attributable to clustering-guided synthetic augmentation.

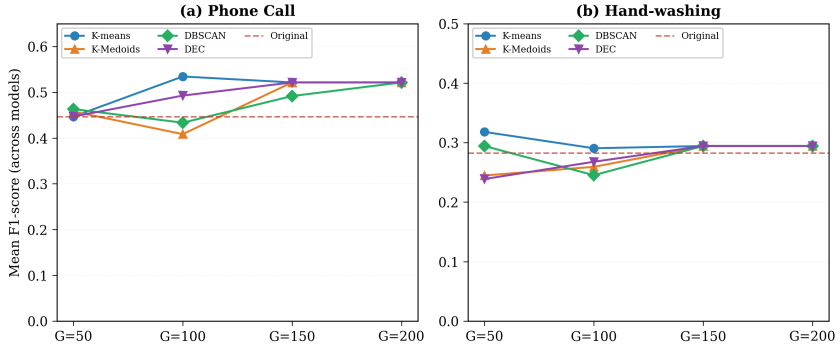

Figure C2: Kyoto7 mean F1-score across detection models for each clustering method and generation volume G. The dashed red line indicates the original baseline. (a) phone call and (b) hand-washing anomaly detection. Each line represents one clustering method, traced from $G = 50$ to $G = 200$.

Figure C1 makes the effectiveness gap between the two activities visible at a glance. three bars standing well above the 0.444 baseline (EE), a 50.2% relative jump. K-Medoids achieves a more conservative 0.571 (+14.2% over its OCSVM baseline of 0.500), its bar noticeably shorter than the other three. For hand-washing (b), the picture flattens: all methods converge to approximately 0.516, a modest +8.4% over the 0.476 baseline. The bars sit nearly at the same height regardless of which clustering backbone drove the generation. This contrast drives home a key finding—when the underlying behavior is flexible and high-variance, augmentation has room to inject diversity and the clustering choice matters. When the behavior is rigid, the ceiling is lower and every method hits it equally.

C.2 Mean F1 Across Generation Volumes

Figure C2 examines how the mean F1-score varies across different generation volumes G for each clustering method. Each line traces one clustering method, while the dashed baseline line provides a reference for the gain attributable to augmentation.

Figure C2 reveals two distinct patterns. For phone call detection (a), performance generally stabilizes at $G \geq 100$. DBSCAN stands out at the low end: at $G = 50$, its mean F1 already surpasses the other three methods, capitalizing on its ability to identify natural groupings without requiring a pre-specified cluster count. Once G reaches 100, all methods converge to a similar plateau, and pushing further to 150 or 200 adds no measurable benefit. For hand-washing (b), the improvement is more gradual. All four lines creep upward as G increases, then flatten—the gap between the

Table C1: Kyoto7 hand-washing anomaly detection results. For each clustering method and generation volume G, we report accuracy, recall, and F1-score for four detection models. "Original" denotes results without augmentation. Bold entries highlight the best score within each clustering block for each metric.

Clustering	G	**OCSVM**			**iForest**			**LOF**			**EE**		
		Acc	Rec	F1	Acc	Rec	F1	Acc	Rec	F1	Acc	Rec	F1
K-means	50	0.525	0.444	0.296	0.550	1.000	0.500	0.725	0.000	0.000	0.725	0.556	0.476
	100	0.550	**0.444**	**0.308**	**0.600**	0.333	0.273	0.575	**0.111**	**0.105**	0.725	0.556	0.476
	150	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
	200	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
K-Medoids	50	0.550	0.333	0.250	**0.550**	0.444	**0.308**	0.650	0.000	0.000	0.725	0.444	0.421
	100	**0.575**	**0.444**	**0.320**	**0.600**	0.333	0.273	0.500	0.000	0.000	**0.750**	0.444	0.444
	150	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
	200	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
DBSCAN	50	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
	100	0.475	**0.556**	**0.323**	**0.550**	0.222	0.182	0.700	0.000	0.000	0.725	0.556	0.476
	150	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
	200	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
DEC	50	**0.575**	**0.444**	**0.320**	**0.575**	0.222	0.191	0.650	0.000	0.000	**0.750**	0.444	0.444
	100	**0.600**	0.333	**0.273**	**0.600**	0.222	0.200	0.625	0.000	0.000	0.725	0.444	0.421
	150	**0.600**	0.333	**0.333**	**0.575**	**0.556**	**0.370**	0.625	0.000	0.000	0.725	0.444	0.421
	200	**0.575**	0.333	**0.261**	0.475	**0.778**	**0.400**	0.675	0.000	0.000	0.625	**0.889**	0.516
Original	–	0.550	0.333	0.250	0.525	0.444	0.296	0.700	0.000	0.000	0.725	0.556	0.476

dashed baseline and the augmented methods stays narrower than for phone call throughout. The visual distance between the baseline line and the augmented lines directly quantifies how much augmentation helps: a wide gap for phone call, a narrow one for hand-washing. The takeaway is clear: high-variance routines absorb synthetic diversity well, while rigid sequences saturate quickly.

C.3 Hand-Washing Detection Results

Table C1 reports the complete hand-washing results for Kyoto7. The layout mirrors the phone call table in the main paper: each row block is a clustering method, columns span four detectors, and "Original" gives the unaugmented baseline. Bold values within each clustering block mark the best score for a given detector-metric combination.

Hand-washing is more structured than phone call, so there is less headroom for augmentation to exploit. EE still reaches its best F1 of 0.516 with K-means, K-Medoids, DBSCAN, and DEC at $G = 150$ and $G = 200$, up from 0.476 without augmentation. OCSVM improves modestly—under DEC at $G = 150$ it hits F1 = 0.333 versus the baseline 0.250, a 33% relative gain within a single detector. LOF returns zero recall across every configuration. Its density-based scoring relies on local neighborhood structure, and the augmented hand-washing data apparently distorts those local densities enough to collapse detection entirely. iForest tells a different story: under DBSCAN at $G = 100$ it reaches its peak F1 of 0.323, the only configuration where iForest meaningfully exceeds the 0.296 baseline. Gains plateau once G reaches 150—the EE column shows 0.516 repeated at $G = 150$ and $G = 200$ across all four clustering methods, a clear signal that the behavioral sub-manifold is saturated. Generating more sequences at that point adds computational cost without moving the needle.

C.4 Clustering Method Comparison

Table C2 summarizes the best F1 per clustering method for both Kyoto7 activities, with percentage change over the original baseline. This table condenses the full per-volume results into a single head-to-head view.

Table C2: Kyoto7 best F1 scores per clustering method. Phone/Handwash F1 report the best per-method F1 across the four detectors; Δ is relative to that detector's own Original (EE 0.444 for K-means/DBSCAN/DEC; OCSVM 0.500 for K-Medoids). "Original" denotes detection without any synthetic augmentation.

Method	**Phone F1**	**Δ (%)**	**Handwash F1**	**Δ (%)**
Original	0.444	–	0.476	–
K-means	0.667	+50.2	0.516	+8.4
K-Medoids	0.571	+14.2	0.516	+8.4
DBSCAN	0.667	+50.2	0.516	+8.4
DEC	0.667	+50.2	0.516	+8.4

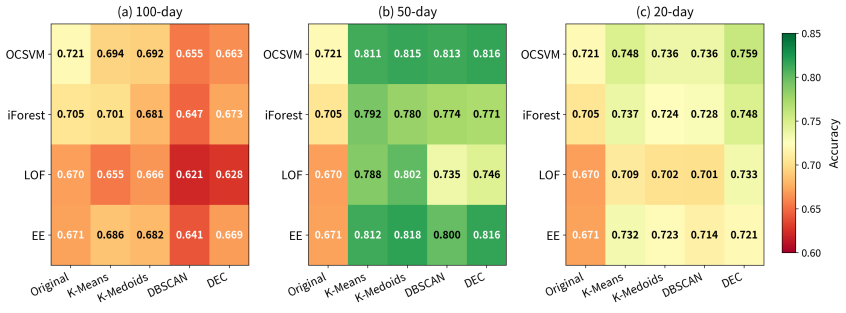

Figure D1: HH111 accuracy heatmap across detection models and clustering methods for (a) 100-day, (b) 50-day, and (c) 20-day augmentation settings. Cell values indicate accuracy; darker colors represent higher performance. The three panels share the same color scale, enabling direct visual comparison across volumes.

All four methods beat the baseline on both activities, but the margin tells a story of its own. For phone call, K-means, DBSCAN, and DEC tie at +50.2% (EE baseline 0.444), while K-Medoids lags at +14.2% (OCSVM baseline 0.500). The gap traces back to how each method conditions the LLM: K-means produces well-separated centroids that translate into clean cluster profiles, DBSCAN captures natural density groupings without a preset K, and DEC learns non-linear latent structure. K-Medoids, by contrast, anchors on actual data points (medoids)—useful for interpretability, but apparently less effective at abstracting the behavioral variation the LLM needs to generate diverse sequences. For hand-washing the picture flattens: every method lands at +8.4%. Kyoto7's multi-resident composition gives clustering rich behavioral diversity to exploit on phone call, but hand-washing's tighter structure caps the achievable gain regardless of which backbone drives the prompt. The practical implication: on diverse data, the choice of clustering method matters; on rigid data, it does not.

D HH111 Additional Results

D.1 Accuracy Heatmap

Figure D1 presents an accuracy heatmap across all detection models, clustering methods, and augmentation volumes. Each cell encodes one model-clustering-volume combination, with darker shades signaling higher accuracy. This visualization compresses the full experimental grid into a single image, making cross-volume trends immediately legible.

Figure D1 reveals a clear volume-dependent arc. The 50-day panel (b) has the darkest cells across nearly all model-clustering combinations, confirming it as the sweet spot. The 20-day panel (c) shows moderate but uneven improvement—some cells brighten over the implicit baseline, others barely change. The 100-day panel (a) tells the most cautionary tale: cells for LOF and DBSCAN fade dramatically, and even OCSVM—the most stable detector—loses ground. The heatmap also exposes a detector hierarchy: OCSVM and EE tend to maintain higher absolute accuracy than iForest and LOF, consistent with their more stable decision boundaries when the training distribution shifts. The

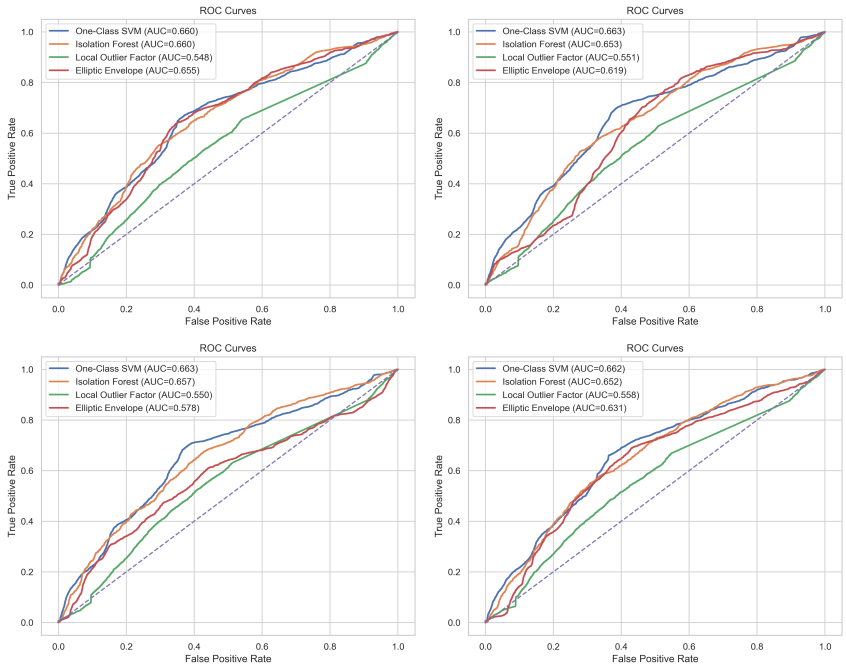

Figure D2: HH111 ROC curves under 20-day augmentation for each clustering method: (a) K-means, (b) K-Medoids, (c) DBSCAN, and (d) DEC.

visual pattern lines up with the numerical tables: augmentation helps up to a point, then the synthetic data overwhelms the genuine signal and accuracy slides backward.

D.2 20-DAY AUGMENTATION

At 20 synthetic days, synthetic data makes up roughly 25% of the training set—the lightest augmentation dose in the HH111 experiments. Figures D2 and D3 present the ROC curves and learning curves for all four detection models under each clustering method at this volume.

Figure D2 shows that OCSVM and Isolation Forest hold the top AUC values (0.55–0.70) across all four clustering methods. LOF and EE, by contrast, hover near the random baseline (AUC $\approx$ 0.50), barely separating anomalous from normal days. The curves themselves look quite similar from one clustering method to the next—the K-means panel differs little from the DBSCAN or DEC panel. Twenty days of synthetic data injects just enough variety to nudge detection above the baseline, but not enough for the choice of clustering backbone to leave a visible fingerprint on discriminative performance.

Figure D3 traces F1-score as a function of training sample size. The lines stay nearly flat across the board—whether a model sees 20% or 100% of the augmented training set, its F1 barely budges. iForest under K-Means shows the most noticeable upward drift, climbing from roughly 0.12 to 0.18 as more data comes in, but even this gain is modest. The flatness tells us something concrete: at 20 synthetic days, the added sequences are too few and too homogeneous to reshape what the detectors have already learned from the real data. The decision boundary barely moves, and performance tags along at baseline levels.

D.3 50-DAY AUGMENTATION

Table D1 reports the complete HH111 results under 50-day augmentation ($D = 50$, $r = 0.3$).

The 50-day setting strikes a better real-to-synthetic balance: augmented data makes up 62.5% of the training set. iForest reaches F1 = 0.274 under K-Means (baseline 0.196, +39.8%), and also clears the baseline under DBSCAN (0.239) and DEC (0.228). LOF, which barely detects anomalies on

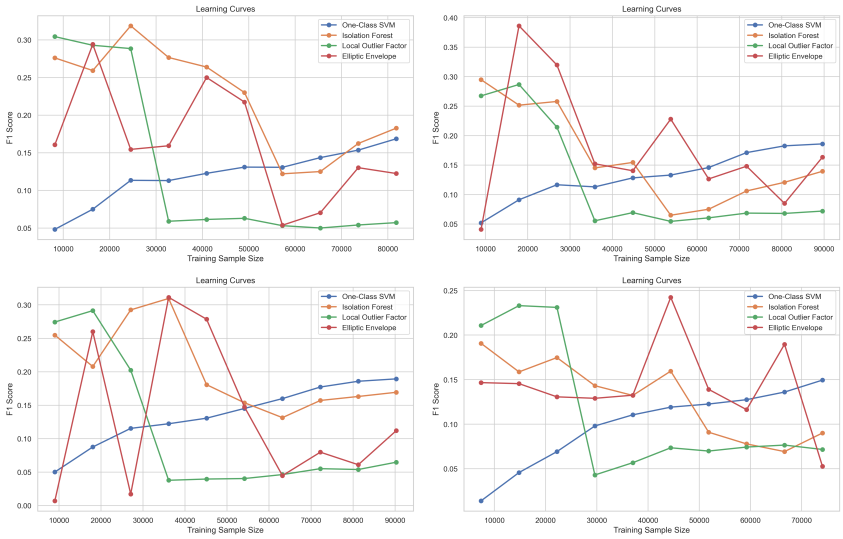

Figure D3: HH111 learning curves under 20-day augmentation for each clustering method: (a) K-means, (b) K-Medoids, (c) DBSCAN, and (d) DEC.

Table D1: HH111 anomaly detection results with 50-day augmentation ($D = 50$, $r = 0.3$). Each row shows a detection model and metric, with columns for the original baseline and four clustering-guided augmentation methods.

Model	Metric	Orig	K-means	K-Medoids	DBSCAN	DEC
OCSVM	Acc	0.721	**0.811**	**0.815**	**0.813**	**0.816**
	Prec	0.661	0.445	0.454	0.454	0.458
	Rec	0.167	0.060	0.088	0.068	0.064
	F1	0.267	0.106	0.147	0.119	0.112
iForest	Acc	0.705	**0.792**	**0.780**	**0.774**	**0.771**
	Prec	0.577	0.393	0.280	0.316	0.295
	Rec	0.118	**0.210**	**0.132**	**0.193**	**0.186**
	F1	0.196	**0.274**	0.179	**0.239**	**0.228**
LOF	Acc	0.670	**0.788**	**0.802**	**0.735**	**0.746**
	Prec	0.289	0.235	0.164	0.241	0.237
	Rec	0.058	0.025	0.022	**0.205**	**0.179**
	F1	0.097	0.042	0.038	**0.221**	**0.204**
EE	Acc	0.671	**0.812**	**0.818**	**0.800**	**0.816**
	Prec	0.254	**0.265**	0.000	**0.312**	0.213
	Rec	0.042	0.004	0.000	**0.071**	0.004
	F1	0.072	0.008	0.000	**0.116**	0.007

the original data (F1 = 0.097), jumps to 0.221 under DBSCAN and 0.204 under DEC. Accuracy climbs for every method, with K-Medoids reaching 0.818 (baseline 0.721, +9.7%). At this volume the synthetic data adds enough behavioral variety to enrich the training distribution without drowning out the genuine signal.

Figure D4 marks a turning point. Unlike the near-identical panels at 20 days, the ROC curves now fan out across clustering methods. iForest under K-Means reaches the highest AUC (0.657), matching its strong F1 showing in Table D1. OCSVM follows closely, its curve bowing further from the diagonal under K-Medoids (AUC 0.632) and DEC (AUC 0.628) than under DBSCAN. The gap between the augmented curves and the original baseline widens visibly—a contrast with the 20-day panels, where augmented and baseline curves sat nearly on top of each other. Fifty days of synthetic data gives the models enough behavioral variety to draw sharper boundaries, and the clustering backbone starts to matter.

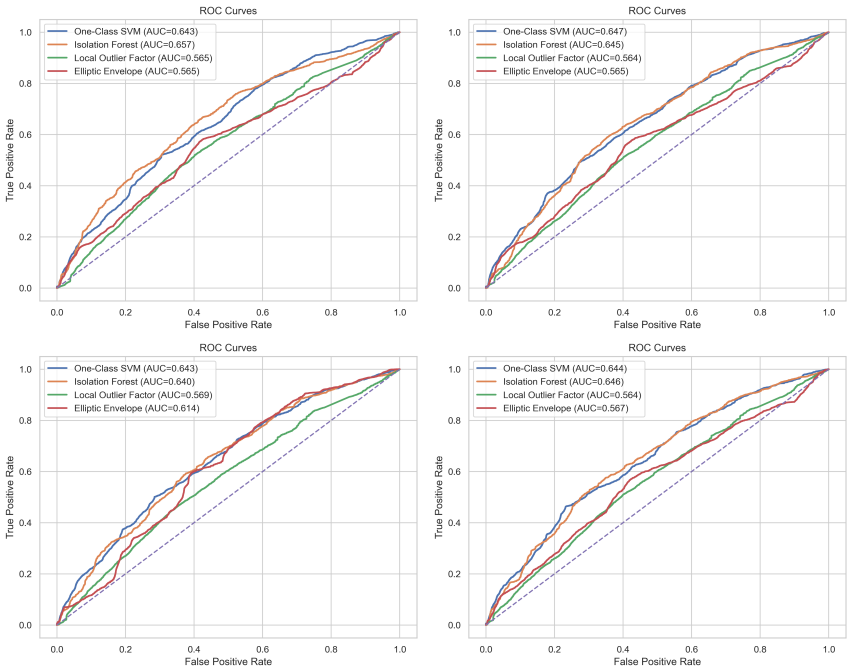

Figure D4: HH111 ROC curves under 50-day augmentation for each clustering method: (a) K-means, (b) K-Medoids, (c) DBSCAN, and (d) DEC. Each subplot shows the ROC curves of four detection models with their respective AUC values.

Figure D5 tells a richer story than its 20-day counterpart. The curves now have genuine slope and shape. iForest under K-Means climbs steadily from F1 ≈ 0.15 at 20% training data to 0.27 at full volume, tracing a smooth upward arc. LOF and EE, in contrast, swing wildly—they spike at 40–60% of the training set and then fall back, a pattern absent at 20 days. Several models peak at an intermediate fraction rather than at full volume, then drift downward as more augmented samples enter. This non-monotonic dip is the first visual hint that synthetic data can overstay its welcome: beyond a certain fraction, the added sequences start pulling the decision boundary away from real behavior rather than enriching it. The 50-day ratio hits a productive middle ground—enough variety to sharpen detection, not so much that the synthetic distribution drowns out genuine patterns.

D.4 100-DAY AUGMENTATION

Table D2 presents the complete anomaly detection results under 100-day augmentation ($D = 100$, $r = 0.3$). The degradation traces to training set composition. The original HH111 dataset holds only 61 days of real sensor data. Under 100-day augmentation, the training set expands to 120 days (20 original plus 100 synthetic), pushing the synthetic share past 83%. Models trained on this set spend most of their capacity fitting LLM-generated patterns rather than the resident's actual behavior. When tested on real data, generalization slips.

Recall degrades the most across all models. OCSVM recall drops from 0.167 to 0.054–0.058; iForest recall drops from 0.118 to 0.034–0.105; LOF recall stays near zero (0.045–0.058). The models become overly conservative, identifying far fewer true anomalies. Some models achieve high precision at the cost of extremely low recall—for instance, EE with DEC reaches 0.697 precision but only 0.086 recall, correctly classifying the few anomalies it detects while missing most.

Accuracy also declines. The original baseline ranges from 0.670 to 0.721 across models; under 100-day augmentation, most drop to 0.621–0.701. The effect is clearest under DBSCAN, where all four models lose accuracy (OCSVM: 0.655, iForest: 0.647, LOF: 0.621, EE: 0.641). The large volume of synthetic data introduces distributional shifts that blur the normal behavior boundary.

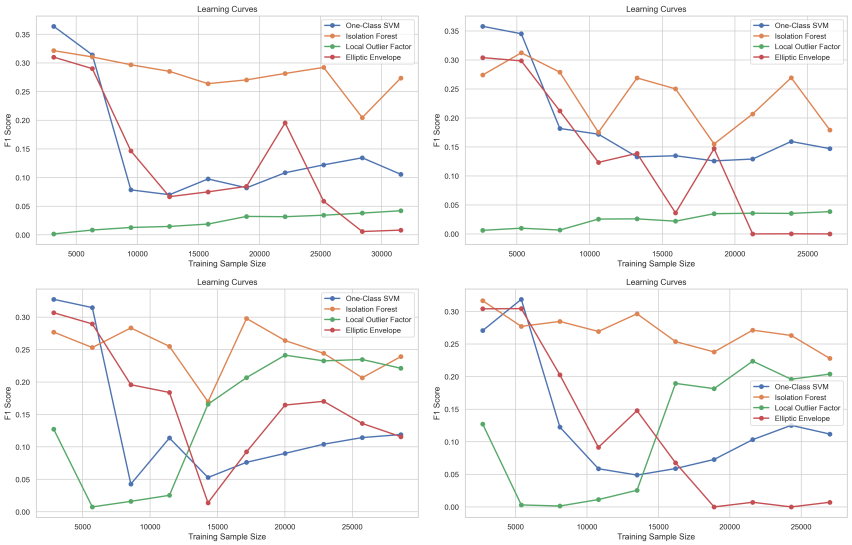

Figure D5: HH111 learning curves under 50-day augmentation for each clustering method: (a) K-means, (b) K-Medoids, (c) DBSCAN, and (d) DEC. Each subplot shows F1-score as a function of training sample size for four detection models.

Table D2: HH111 anomaly detection results with 100-day augmentation ($D = 100$, $r = 0.3$). Each row shows a detection model and metric, with columns for the original baseline and four clustering-guided augmentation methods.

Model	Metric	Orig	K-means	K-Medoids	DBSCAN	DEC
OCSVM	Acc	0.721	0.694	0.692	0.655	0.663
	Prec	0.661	**0.662**	**0.673**	**0.702**	**0.684**
	Rec	0.167	0.054	0.056	0.058	0.056
	F1	0.267	0.099	0.104	0.107	0.104
iForest	Acc	0.705	0.701	0.681	0.647	0.673
	Prec	0.577	**0.655**	0.462	0.561	**0.705**
	Rec	0.118	0.093	0.034	0.052	0.105
	F1	0.196	0.163	0.064	0.095	0.183
LOF	Acc	0.670	0.655	0.666	0.621	0.628
	Prec	0.289	0.243	**0.320**	**0.328**	**0.290**
	Rec	0.058	0.046	0.049	0.058	0.045
	F1	0.097	0.078	0.085	**0.098**	0.082
EE	Acc	0.671	**0.686**	**0.682**	0.641	0.669
	Prec	0.254	**0.413**	**0.448**	**0.477**	**0.697**
	Rec	0.042	0.004	0.013	**0.044**	**0.086**
	F1	0.072	0.007	0.026	**0.081**	**0.154**

The ROC curves in Figure D6 paint a picture of retreat. AUC values drop across the board compared to the 20-day and 50-day panels. OCSVM, which held 0.60–0.65 at 20 days, slips to 0.55–0.60—its curve pulling back toward the diagonal. iForest suffers a steeper decline, particularly under DBSCAN where AUC falls to 0.52. The K-Medoids panel shows the least damage, with OCSVM retaining an AUC of 0.596, but even this is below its 50-day peak. When 83% of the training set is synthetic, the models spend more capacity fitting LLM-generated patterns than learning the resident's actual behavior. The result is a set of ROC curves that hug the diagonal—discriminative power eroded by an oversupply of artificial data.

The learning curves in Figure D7 sharpen the cautionary picture. Performance peaks at small training fractions—often below 30%—then slides downhill as more augmented data enters the mix. This is the opposite of what happened at 50 days, where several models climbed toward an intermediate

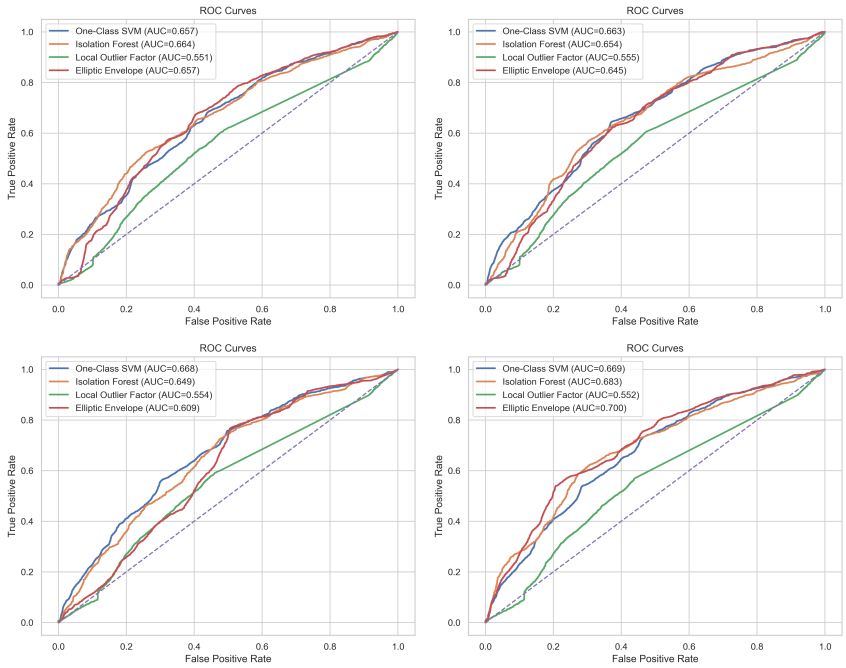

Figure D6: HH111 ROC curves under 100-day augmentation for each clustering method: (a) K-means, (b) K-Medoids, (c) DBSCAN, and (d) DEC. Each subplot shows the ROC curves of four detection models with their respective AUC values.

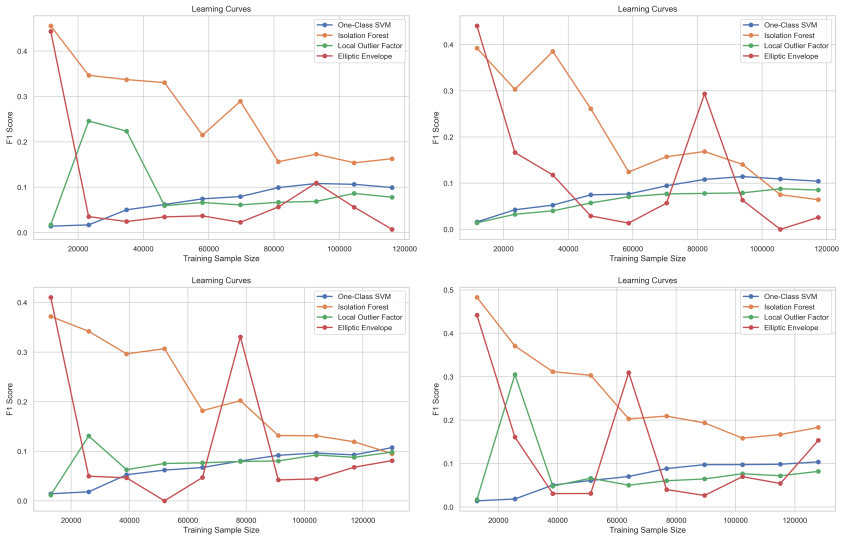

Figure D7: HH111 learning curves under 100-day augmentation for each clustering method: (a) K-means, (b) K-Medoids, (c) DBSCAN, and (d) DEC. Each subplot shows F1-score as a function of training sample size for four detection models.

peak. The decline is steepest for LOF and EE: their F1 scores drop to near-baseline levels by the time the full training set is used. OCSVM and iForest are more resilient but still lose ground. The curves confirm that beyond a certain threshold, synthetic samples stop adding signal and start adding noise. The model's decision boundary, stretched to accommodate an artificial distribution, drifts away from the real behavior it needs to capture.

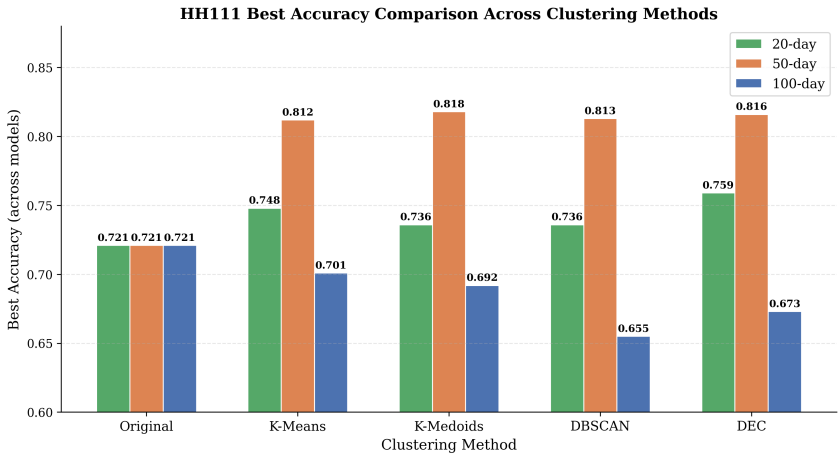

Figure D8: HH111 best accuracy comparison across 20-day, 50-day, and 100-day augmentation settings for each clustering method. The dashed horizontal line indicates the original baseline.

Table D3 distills the best scores across all three volumes into a single compact view. Reading left to right, the 20-day column shows modest gains, the 50-day column peaks, and the 100-day column retreats—an inverted-U pattern that holds for accuracy, F1, and recall alike. Precision is the lone exception: it climbs at 100 days (0.702 with DBSCAN), but only because the models become so conservative that they flag almost nothing, catching the few anomalies they do identify with high confidence while missing the rest.

Table D3: HH111 best accuracy and F1 comparison across augmentation volumes. Values represent the best score achieved across all detection models and clustering methods.

Metric	20-day	50-day	100-day
Best Acc.	0.759 (DEC)	0.818 (KM)	0.701 (KM)
Best F1	0.189 (DB)	0.274 (KM)	0.183 (DEC)
Best Rec.	0.115 (iF)	0.210 (iF)	0.105 (EE)
Best Prec.	0.604 (DB)	0.458 (DEC)	0.702 (DB)

The 50-day column sweeps the board: highest accuracy (0.818, K-Medoids), highest F1 (0.274, K-Means), and highest recall (0.210, iForest). The 100-day column trails on every metric except precision. That precision spike (0.702, DBSCAN) comes at a steep price—recall drops to 0.058, meaning the model catches fewer than one in ten true anomalies. In a safety-critical smart home, a missed fall or wandering episode carries far higher cost than a false alarm. The numbers make the deployment lesson clear: 50 synthetic days is the practical ceiling for HH111, and pushing beyond it trades genuine detection for a false sense of precision.

Figure D8 visualizes the cross-volume accuracy pattern. Across all four clustering methods, the 50-day bars exceed both the 20-day and 100-day values, and the baseline sits below every 20-day and 50-day result. The 100-day bars, by contrast, fall back toward or below the baseline for most methods—K-means drops from 0.811 to 0.701, and DBSCAN suffers the steepest decline to 0.655. The bars trace an inverted U: more synthetic data helps up to a point, then actively harms.

D.5 Cross-Volume Summary

Table D4 consolidates the best F1 per clustering method across all three augmentation volumes, offering a head-to-head view of how volume shifts detection quality.

The F1 columns tell a clear story: only K-Means at 50 days (0.274) clears the original baseline of 0.267. Every other clustering–volume combination falls short. The 50-day column also produces the highest F1 for DBSCAN (0.239) and DEC (0.228), though neither reaches the baseline. At 100 days, F1 collapses across the board—K-Medoids drops to 0.104, DBSCAN to 0.107. The 20-day results

Table D4: HH111 best F1 scores per clustering method, showing the maximum F1 achieved across all detection models for 20-day, 50-day, and 100-day augmentation settings.

Method	20-day F1	50-day F1	100-day F1
Original	0.267	0.267	0.267
K-Means	0.183	**0.274**	0.163
K-Medoids	0.186	0.179	0.104
DBSCAN	0.189	0.239	0.107
DEC	0.149	0.228	0.183

are modest but more even-handed: K-Medoids (0.186) and DBSCAN (0.189) edge past their 50-day counterparts, suggesting that for these two methods, a smaller synthetic dose works better. K-Means and DEC, on the other hand, thrive at 50 days. The takeaway: no single volume suits every clustering backbone, but 50 days gives the best shot at beating the baseline.

Table D5 presents the corresponding best accuracy per clustering method. Unlike F1, accuracy improves at every volume for every method except 100-day, where it falls back.

Table D5: HH111 best accuracy per clustering method, showing the maximum accuracy achieved across all detection models for 20-day, 50-day, and 100-day augmentation settings.

Method	20-day Acc	50-day Acc	100-day Acc
Original	0.721	0.721	0.721
K-Means	**0.748**	**0.811**	0.701
K-Medoids	**0.736**	**0.818**	0.692
DBSCAN	**0.736**	**0.813**	0.655
DEC	**0.759**	**0.816**	0.673

Accuracy paints a more optimistic picture than F1. At 50 days, every clustering method pushes accuracy above 0.81—K-Medoids reaches 0.818, DEC 0.816, DBSCAN 0.813, and K-Means 0.811, all up from the 0.721 baseline. The 20-day column also beats the baseline across the board, though by a slimmer margin (0.736–0.759). The 100-day column reverses these gains: every method drops below the original 0.721, with DBSCAN suffering the steepest fall to 0.655. The accuracy improvement at 20 and 50 days comes mainly from the models learning to classify the majority class (normal days) more confidently after seeing more varied training data. F1, which weights precision and recall together, reveals that this accuracy gain does not always translate into better anomaly catching—a gap that Tables D4 and D5 together expose.

Table D6 places F1 and accuracy side by side for the two most informative volumes—20-day and 50-day—dropping the 100-day column where performance collapsed.

Table D6: HH111 best F1 and accuracy per clustering method across 20-day and 50-day augmentation settings.

	Best F1		Best Acc	
Method	20-day	50-day	20-day	50-day
Original	0.267	0.267	0.721	0.721
K-Means	0.183	**0.274**	**0.748**	**0.811**
K-Medoids	0.186	0.179	**0.736**	**0.818**
DBSCAN	0.189	0.239	**0.736**	**0.813**
DEC	0.149	0.228	**0.759**	**0.816**

Table D6 makes the tension between accuracy and F1 visible. Accuracy climbs at both 20 and 50 days for every method. F1, though, only surpasses the baseline at 50 days with K-Means (0.274 vs. 0.267)—a razor-thin margin. The other three clustering methods stay below 0.267 even at the 50-day sweet spot. K-Medoids posts the highest single accuracy (0.818), but its F1 actually drops from 0.186 at 20 days to 0.179 at 50 days, revealing that the accuracy gain comes from better normal-day classification, not from catching more anomalies. The gap between 20-day and 50-day F1

for K-Means (0.183 $\rightarrow$ 0.274) is the widest in the table—volume matters more than clustering choice once the right backbone is in place. At 100 days (Tables D4 and D5), both metrics collapse, with DBSCAN falling hardest (accuracy 0.655, F1 0.107). Deploying the framework on single-resident data thus requires joint tuning of clustering method and augmentation volume—neither alone is enough.