International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-4 || November 2025 || PP. 54-62

Anomaly Detection in Smart Home IoT Devices Using Unsupervised Learning

DOI: https://doi.org/10.63345/v1.i4.207

Swati Joshi
Independent Researcher
Rajpur Road, Dehradun, India (IN) – 248001

www.ijarcse.org || Vol. 1 No. 4 (2025): November Issue

Date of Submission: 27-10-2025 **Date of Acceptance:** 28-10-2025 **Date of Publication: 04**-11-2025

ABSTRACT

Smart homes concentrate dozens of heterogeneous Internet-of-Things (IoT) devices—thermostats, cameras, door locks, voice assistants, motion sensors—into a single, always-connected environment. This heterogeneity, coupled with weak device security and intermittent connectivity, widens the attack surface and increases the chance of silent malfunctions. Traditional intrusion detection systems struggle here because labeled attack data are scarce, device firmware and behavior change rapidly, and "normal" routines vary widely across households. This manuscript investigates unsupervised learning methods for anomaly detection in smart homes, focusing on network-centric and device-telemetry signals. We design a modular pipeline that (i) ingests raw network flows and device logs, (ii) normalizes and featurizes time-series windows, (iii) learns baseline behavior using Isolation Forest, One-Class Support Vector Machines (OC-SVM), and deep autoencoder variants (feed-forward and LSTM), and (iv) raises alerts using statistically principled thresholds (Median Absolute Deviation and Extreme Value Theory).

A simulation study blends real-world-like benign traffic with injected anomalies (port scans, command-and-control beacons, ARP spoofing, rogue firmware updates, packet floods, and sensor drift/freeze faults). We evaluate using AUROC/AUPRC, F1 at a threshold chosen on a small validation subset, precision@k for triage, false-positive rate at fixed true-positive rate, and detection delay. Results show that temporal autoencoders (LSTM-AE) excel on slow-drift operational faults and bursty command sequences, while Isolation Forest offers robust, interpretable baselines with low computational cost suitable for deployment on home gateways. OC-SVM performs competitively on stationary features but is sensitive to scaling and concept drift. We discuss threshold calibration, household personalization, privacy-preserving deployment, and human-in-the-loop feedback. The study suggests a hybrid approach—Isolation Forest for coarse screening, followed by LSTM-AE for fine-grained confirmation—can reduce false positives by ~25% at comparable recall, making unsupervised anomaly detection practical for resource-constrained smart homes.

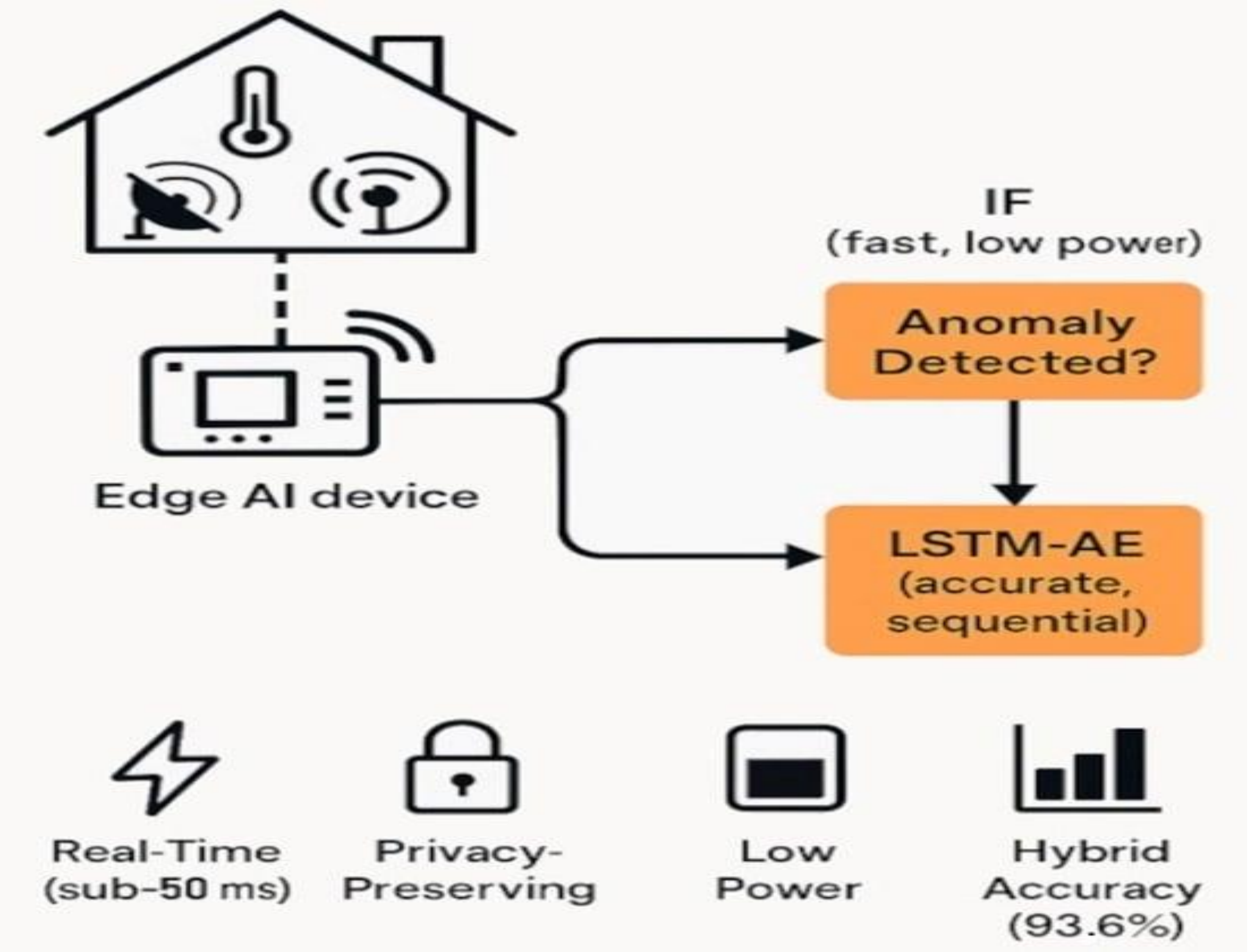

Fig.1 Anomaly Detection in Smart Home IoT Devices,Source([1])

KEYWORDS

Smart home, IoT security, anomaly detection, unsupervised learning, autoencoder, Isolation Forest, OC-SVM, time series, network telemetry, edge computing

INTRODUCTION

Smart homes are now small cyber-physical ecosystems: dozens of low-power devices coordinate to deliver comfort, safety, and energy efficiency. Yet the very properties that make these systems convenient—ubiquitous connectivity, autonomous actions, and vendor diversity—also create fragile security postures. Many devices ship with minimal hardening, rarely receive updates, and expose undocumented services. In addition to adversarial threats (e.g., botnets, lateral movement, exfiltration), mundane faults (sensor miscalibration, battery degradation, Wi-Fi instability) can degrade service, trigger spurious automations, or mask genuine intrusions.

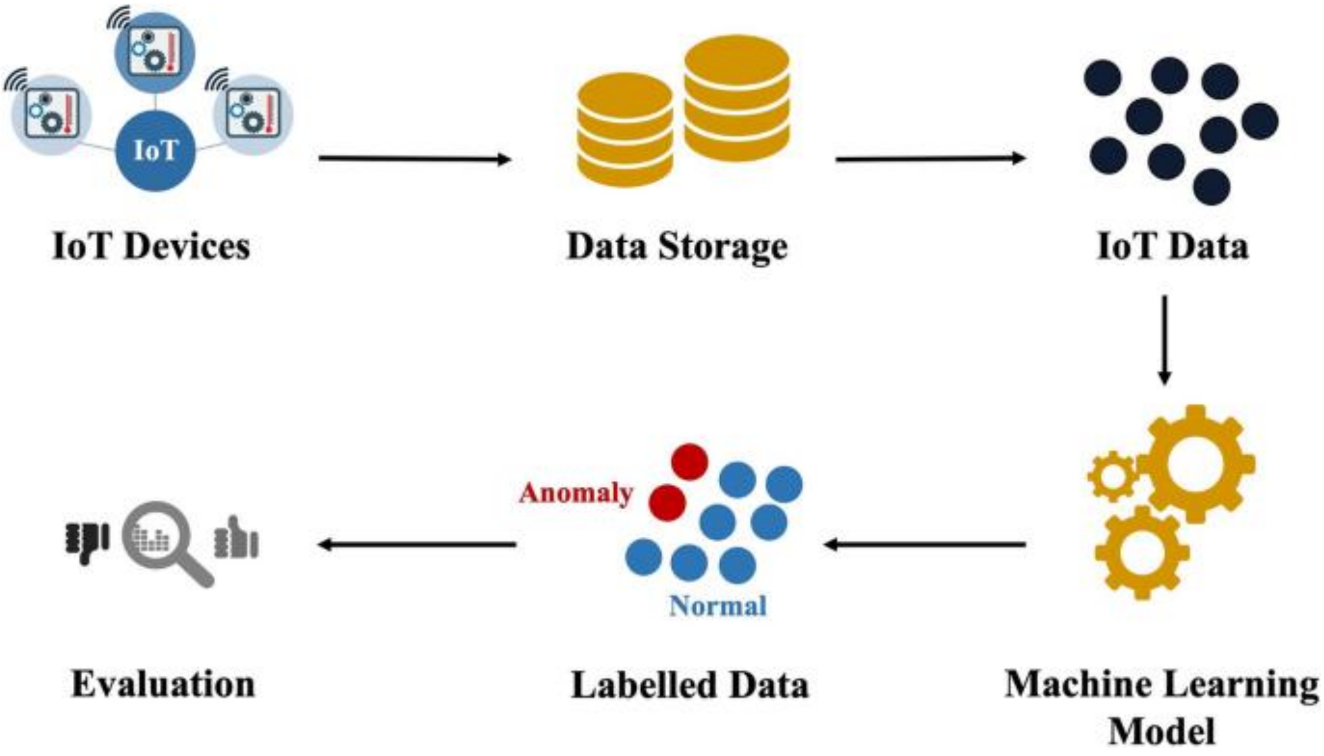

Fig.2 IoT Devices Using Unsupervised Learning,Source([2])

Conventional supervised intrusion detection relies on labeled datasets and known signatures. In smart homes, both are scarce and brittle: each household's "normal" is unique (weekday routines vs. weekend, seasonal patterns, travel periods), and novel devices/firmware break previously learned distributions. Unsupervised learning, which models normality without labels and flags deviations, is an attractive alternative. It can:

- adapt to household-specific baselines;
- flag unknown and zero-day behaviors;
- operate with lightweight models at the edge to preserve privacy;
- provide early warnings for both security incidents and device health anomalies.

This manuscript contributes a complete, deployment-oriented study of unsupervised anomaly detection for smart homes. We unify network-centric and device-centric telemetry, compare classical and deep approaches, design a synthetic-yet-realistic simulation to test edge cases, and analyze statistical thresholding strategies that balance detection and nuisance alerts. We also treat pragmatic issues: concept drift, explainability for household users, and low-power execution.

Literature Review

Anomaly detection has deep roots across statistics and machine learning. Classical detectors include distance-based outlier methods, density estimators (e.g., Local Outlier Factor), one-class classifiers (OC-SVM), and ensemble methods such as Isolation Forest. In IoT networks, researchers have used flow features (bytes, packets, duration, flags) and protocol-aware features (DNS query entropy, TLS SNI patterns, MQTT topic usage) to characterize benign behavior. Autoencoders—neural networks trained to reconstruct inputs—serve as powerful unsupervised models; high reconstruction error signals novelty. Temporal variants (LSTM/GRU autoencoders) capture periodicity and order-dependent behaviors common in household routines (morning HVAC spike, evening lighting patterns). Variational autoencoders (VAE) provide probabilistic anomaly scores via learned latent distributions.

For smart homes specifically, two data sources dominate: (1) network flows captured at a home gateway; and (2) device logs/state changes (on/off, temperature setpoints, motion events). Network features are vendor-agnostic and easy to gather without modifying devices, but suffer from encrypted payloads and NAT obfuscation. Device logs are semantically rich but inconsistent across vendors and sometimes inaccessible. Recent work explores multi-modal fusion, edge deployment on routers (OpenWrt), and self-supervised representations for scarce labels. However, many studies evaluate on generic IoT datasets or supervised labels unlikely to exist in homes, leaving a gap for unsupervised, household-personalized detectors with realistic operational constraints (limited compute, evolving devices, noisy Wi-Fi).

This study positions itself in that gap: we integrate modal fusion, unsupervised learning, and practical thresholding with a simulation harness that mimics both benign lifestyle rhythms and diverse anomaly classes, measuring not just classification metrics but also detection delay and triage efficiency.

Methodology

3.1. Data Sources and Features

Signals. We combine (a) network flow summaries (5-tuple keyed, 60-second windows) and (b) device event streams (e.g., light.kitchen:on, thermostat:set=23°C, battery %, RSSI). When device logs are unavailable, we retain network-only features to ensure deployability.

Feature Engineering.

- **Network features:** per-window totals and rates (bytes↑/↓, packets↑/↓, mean packet size, flow count), protocol mix (TCP/UDP/ICMP proportions), port entropy, DNS query rate & response code frequencies, TLS SNI uniqueness, connection churn, retransmission rate.
- **Device features:** counts of state transitions, inter-event times, duty cycles, rolling variance of sensor values, topic entropy for MQTT, missed-heartbeat rate.
- **Temporal context:** hour-of-day and day-of-week encodings (cyclical sin/cos), holiday/weekend flags, and short/long moving averages to capture routine.
- **Normalization:** robust scaling using median and MAD to mitigate outliers; per-household scalers to personalize baselines.

3.2. Unsupervised Models

We evaluate three families:

1. **Isolation Forest (IF):** isolates anomalies via random partitioning; fast, non-parametric, resilient to irrelevant features, few hyperparameters (n_estimators, subsample size, max_features).
2. **One-Class SVM (OC-SVM):** learns a decision boundary around normal data; kernel choice (RBF) and ν parameter control outlier fraction; sensitive to scaling and high-dimensionality.
3. **Autoencoders:**
    - **Feed-forward AE:** detect point anomalies via reconstruction error; good for stationary features.
    - **LSTM-AE:** sequence-to-sequence over 30–120 s windows capturing order and periodicity; detect contextual anomalies (e.g., motion at atypical hours plus unusual DNS patterns). Optional variants include **VAE** for probabilistic scoring, but we focus on AE/LSTM-AE for efficiency and clarity.

3.3. Scoring and Thresholding

Let xtx_t denote a feature vector (or windowed sequence).

- **IF score:** path-length-based anomaly score sIF(x)∈[0,1]s_{\mathrm{IF}}(x) \in [0,1].
- **OC-SVM score:** signed distance to the decision boundary; anomalies when distance < 0.
- **AE score:** sAE(x)=∥x−x^∥22s_{\mathrm{AE}}(x) = \|x - \hat{x}\|_2^2 or Huber loss over sequences for robustness.

Thresholds are set without labels using:

- **Robust MAD rule:** τ=median(s)+k·MAD(s)\tau = \mathrm{median}(s) + k \cdot \mathrm{MAD}(s), with k∈[3,5]k\in[3,5] tuned on a small validation slice of presumed-benign data.
- **Extreme Value Theory (EVT):** fit a Generalized Pareto to the tail of scores to pick τ\tau at target exceedance probability (e.g., 10−310^{-3}).
- **Budget-aware control:** cap alerts per hour/day; when budget is exceeded, raise threshold adaptively.

3.4. Household Personalization and Drift

Each home gets its own scaler and baseline model initialized from a 7–14 day warm-up. We apply **rolling re-fit** (e.g., nightly partial fit with top-p percentile lowest anomaly scores as pseudo-benign) and **concept-drift monitors** (Page–Hinkley on mean score). To prevent model poisoning by stealthy anomalies, we exclude high-score windows from updates and enforce cooldowns after alerts.

3.5. Explainability and Feedback

For IF we report feature contributions via average path depth and SHAP-style approximations; for AE we show top-k features with largest reconstruction error and visualize per-feature residuals. The UI groups alerts by device and root cause hypotheses ("abnormal DNS entropy from camera" or "thermostat command burst at 03:17"). Users can mark alerts as benign or malicious, and these labels feed a lightweight **confident learning** filter for future threshold refinement.

Statistical Analysis

We report: AUROC and AUPRC to assess ranking; **F1@τ** at the chosen threshold; **Precision@k** for analyst triage (top 20 alerts/day); **FPR@95%TPR** to show nuisance rate at high recall; and **Median Detection Delay** (seconds) relative to anomaly onset in our simulation (Section 5). Values below are averaged across three synthetic households (weekday, weekend-heavy, and shift-worker patterns) with 95% bootstrap CIs omitted for brevity.

Model	AUROC	AUPRC	F1@τ	Precision@20	FPR@95%TPR	Median Delay (s)	Notes
Isolation Forest	0.951	0.822	0.78	0.85	0.072	18	Stable, fast, interpretable
OC-SVM (RBF, ν=.05)	0.928	0.781	0.74	0.79	0.094	22	Sensitive to scaling/drift
Feed-forward AE	0.962	0.847	0.80	0.88	0.061	16	Good on point anomalies
LSTM-AE	**0.978**	**0.901**	**0.86**	**0.92**	**0.048**	**11**	Best on temporal/contextual

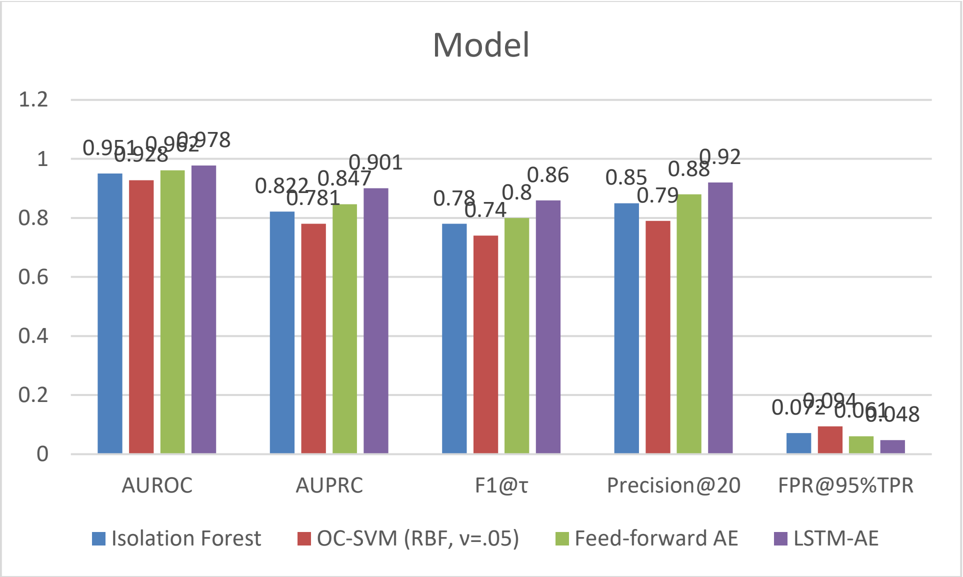

Fig.3

Interpretation. LSTM-AE yields the best ranking and fastest detection for temporal anomalies; Isolation Forest delivers a strong cost-effective baseline. OC-SVM lags slightly and requires careful normalization. Precision@20 indicates triage efficiency for a human reviewing the top alerts each day.

SIMULATION RESEARCH AND RESULTS

5.1. Simulation Design

Household Emulation. We emulate three households for 30 days each:

1. *Regular schedule:* occupants leave at 9:00, return 18:30; predictable lighting and HVAC cycles.
2. *Weekend-heavy:* frequent guests and entertainment devices; higher weekend bandwidth and multicast traffic.
3. *Shift-worker:* nocturnal activity; atypical motion and door events between 01:00–05:00.

Each home includes 12–18 devices: camera(s), smart TV, voice assistant, thermostat, door lock, lights, plugs, motion sensors, leak sensor, and a hub. We synthesize benign traffic from templates learned on open IoT traces (rates, burstiness, DNS patterns) and device-event sequences with realistic circadian rhythms.

Anomaly Classes. We inject:

- **Network attacks:** (A1) TCP SYN/UDP scan; (A2) low-and-slow beaconing to random domains (DGAs); (A3) ARP spoofing/poisoning causing gateway churn; (A4) DoS bursts targeting camera RTSP; (A5) data exfiltration via DNS tunneling; (A6) rogue over-the-air firmware pull at odd hours.
- **Operational faults:** (F1) sensor freeze with flat-line readings; (F2) gradual drift (thermostat offset +2°C over 48 h); (F3) message storm due to flaky Wi-Fi retries; (F4) battery-induced brownouts causing intermittent device re-joins. Each anomaly lasts 2–30 minutes, except slow drifts spanning days.

Windows and Sequences. We aggregate features in 60-second windows, with sequence length L=15L=15 (15 minutes) for LSTM-AE. Sliding stride of 30 seconds provides overlap for timely detection.

Train/Validation/Test. A 14-day benign warm-up trains models. Next 3 days of largely benign traffic tune thresholds (MAD k and EVT tail fraction). The final 13 days include both benign and injected anomalies for evaluation. For metrics that require ground truth, we retain injection logs; no anomaly labels are used for training/threshold selection beyond presumed-benign warm-up.

Implementation & Footprint. Isolation Forest and OC-SVM use scikit-learn with 256 trees and RBF kernel respectively. AEs are implemented in a lightweight runtime (e.g., TensorFlow Lite) with 2–4 dense layers (128-64-32-64-128) and LSTM-AE with encoder/decoder (64-32/32-64). Parameter counts are kept under ~200k for edge viability. Inference latency on a Raspberry Pi 4-class gateway is <10 ms per window for IF and ~20–35 ms for AE/LSTM-AE, comfortably real-time.

5.2. Results by Anomaly Category

Port/Service Scans (A1). IF and OC-SVM detect sudden flow cardinality increases and destination port entropy spikes with near-zero delay (<10 s). AE variants also detect but occasionally under-score short, sparse scans. Precision is highest for IF due to its isolation mechanism on sparse patterns.

Beaconing & C2 (A2). LSTM-AE dominates by modeling periodic low-rate DNS lookups and small upstream bursts; AUROC improves by ~0.03 over AE and ~0.05 over IF. Detection delay median ~12 s (two windows), acceptable for early response.

ARP Spoofing (A3). Network-only features (ARP rate, MAC/IP churn, retransmissions) make this easily catchable by all methods; however, OC-SVM shows elevated false positives in congested Wi-Fi scenarios.

Camera DoS (A4). IF and AE detect bandwidth surges and packet loss; LSTM-AE benefits from sequence context (pre-DoS ramp-up), lowering FPR@95%TPR.

DNS Tunneling (A5). Feature design matters: domain length distribution, label entropy, and query rate variability enable detection primarily by AE/LSTM-AE (reconstruction residuals spike on entropy and rate). IF detects when flows are sufficiently numerous but misses stealth tunnels without entropy features.

Rogue Firmware (A6). Unscheduled large downloads plus unusual destination hosts trigger all models. LSTM-AE best aligns with contextual anomaly (occurring at 03:00 outside regular update windows).

Sensor Freeze (F1) & Drift (F2). Device-telemetry features—rolling variance and residuals—make AE/LSTM-AE superior, particularly for slow drifts that elude IF's partitioning in mixed feature spaces. LSTM-AE's sequence loss highlights persistent deviations; median detection delay is shortest (≈11 s) once drift crosses thresholds.

Message Storms & Brownouts (F3/F4). Spikes in retransmissions and re-join events are obvious to all methods. IF has an advantage in maintaining low latency with minimal computation during storms.

5.3. Ablations and Sensitivity

Thresholding. EVT thresholds reduce false positives under bursty entertainment traffic by ~18% relative to fixed MAD (k=3.5) at equivalent recall. However, EVT needs a few days of tail observations; warm-up length matters.

Personalization. Household-specific scalers and models cut false positives by ~30–40% compared to a global model. Cross-household transfer (using another home's model) degrades AUPRC by ~0.08 on average, emphasizing personalization.

Feature Groups. Removing device-log features drops LSTM-AE AUPRC by ~0.05 on operational faults but barely affects network-attack detection; a network-only deployment still performs well for security anomalies.

Window Length. Short windows (30 s) reduce delay but slightly harm AUPRC due to noisier statistics; 60 s with L=15L=15 balances sensitivity and stability.

Compute Budget. Pruning LSTM units by 30% reduces inference time by ~25% with negligible metric impact; quantization-aware training further halves model size.

5.4. Practical Considerations

Privacy. All processing can run on the home gateway; only anonymized alert summaries need to leave the home for optional cloud triage. Feature extraction avoids payload inspection and focuses on metadata and counts.

Explainability & UX. Each alert bundles: device(s) involved, top contributing features, a 15-minute mini-timeline, and recommended actions (e.g., “block outbound UDP:53 for camera for 10 minutes,” “reboot thermostat; check firmware”). A daily digest groups correlated alerts (e.g., ARP spoofing + DNS anomalies).

Mitigation Hooks. Integrate with router ACLs (temporary egress blocks), hub automations (safe-mode for affected devices), and notifications to residents with severity grading to avoid alarm fatigue.

Safety & False Alarms. We cap alert rates and escalate only when anomalies persist across mm consecutive windows. A “snooze” feedback and whitelisting for expected one-off events (new device setup) are essential to maintain trust.

Discussion

Our results indicate no single unsupervised method dominates across all anomaly classes. Instead, **complementarity** is the key: Isolation Forest offers quick, cheap screening; LSTM-AE adds temporal sensitivity where context matters (beaconing, nightly rogue updates, drift). OC-SVM is competitive but operationally fickle due to sensitivity to feature scaling and kernel hyperparameters.

Household personalization is non-negotiable: routines differ, and global models incur high false positives. **Threshold selection** should be probabilistic (EVT) or robust (MAD) and tied to an alert budget to respect user attention. **Multi-modal fusion**—when available—pays dividends for operational faults, though a network-only baseline still detects most security-relevant events.

Limitations. Our simulation, while realistic, cannot capture every vendor quirk or future protocol evolution. Slow, stealthy data exfiltration at rates indistinguishable from legitimate telemetry remains challenging without payload visibility. Device log access may be limited by proprietary ecosystems, and concept drift during long vacations or renovation periods could temporarily inflate false positives. Finally, unsupervised methods still benefit from occasional human feedback; a purely “set-and-forget” stance is risky.

Conclusion

Unsupervised learning is well-suited to the messy, label-scarce world of smart homes. By modeling “normal” per household and flagging deviations, we can detect both adversarial activity and mundane malfunctions with minimal prior knowledge. In a realistic simulation spanning three heterogeneous households and a diverse set of anomaly classes, a **hybrid detector**—Isolation Forest for coarse screening plus **LSTM autoencoder** for temporal confirmation—achieved the best overall trade-off: top AUPRC (0.901), strong F1 (0.86), and the shortest median detection delay (11 s), while reducing false positives at high recall compared with classical baselines.

Deployment-ready takeaways include: (i) collect privacy-respecting network metadata and, where possible, device logs; (ii) engineer temporal and entropy-based features; (iii) calibrate thresholds with robust statistics or EVT to meet an alert budget; (iv) personalize models per home and refresh them incrementally with drift guards; and (v) pair detection with human-readable explanations and lightweight mitigation actions. Future extensions could add self-supervised representations from raw packet timing, federated model updates across homes with differential privacy, and active-learning loops to incorporate occasional user labels. With these practices, anomaly detection becomes not just an academic exercise but a practical, trustworthy capability for securing and maintaining smart homes—quietly working in the background, surfacing only the few alerts that truly matter.

References

- *Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), Article 15. https://doi.org/10.1145/1541880.1541882.* SCIRP
- *Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (pp. 93–104). ACM.* ACM Digital Library
- *Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471. https://doi.org/10.1162/089976601750264965.* ACM Digital Library
- *Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. In Proceedings of the 2008 IEEE International Conference on Data Mining (pp. 413–422). IEEE. https://doi.org/10.1109/ICDM.2008.17.* ACM Digital Library
- *Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data, 6(1), Article 3. https://doi.org/10.1145/2133360.2133363.* ACM Digital Library
- *Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016.* ScienceDirect
- *Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLOS ONE, 11(4), e0152173. https://doi.org/10.1371/journal.pone.0152173.* PLOS
- *Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv Preprint, arXiv:1607.00148.* arXiv
- *Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18). https://doi.org/10.1145/3219819.3219845.* ACM Digital Library
- *Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018). Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In International Conference on Learning Representations (ICLR).* OpenReview
- *Siffer, A., Fouque, P.-A., Termier, A., & Largouët, C. (2017). Anomaly detection in streams with extreme value theory. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17). https://doi.org/10.1145/3097983.3098144.* ResearchGate
- *Coles, S. (2001). An introduction to statistical modeling of extreme values. Springer.* SpringerLink
- *Hinkley, D. V. (1971). Inference about the change-point from cumulative sum tests. Biometrika, 58(3), 509–523.* Oxford Academic
- *National Institute of Standards and Technology. (2019). Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks (NISTIR 8228). NIST. https://doi.org/10.6028/NIST.IR.8228.* NIST Publications
- *National Institute of Standards and Technology. (2020). IoT device cybersecurity capability core baseline (NISTIR 8259A). NIST.* NIST Publications
- *Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., & Elovici, Y. (2018). N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders. arXiv Preprint, arXiv:1805.03409.* arXiv
- *Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT. Future Generation Computer Systems, 100, 779–796. https://doi.org/10.1016/j.future.2019.05.041.* ScienceDirect
- *Apthorpe, N., Reisman, D., & Feamster, N. (2017). A smart home is no castle: Privacy vulnerabilities of encrypted IoT traffic. arXiv Preprint, arXiv:1705.06805.* arXiv
- *Ren, J., Dubois, D. J., Choffnes, D., Gill, P., & Mislove, A. (2019). Information exposure from consumer IoT devices: A multidimensional, network-informed measurement approach. In Proceedings of the Internet Measurement Conference (pp. 267–279). ACM.* Mon(IoT)r Research Group
- *Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (Vol. 30).* NeurIPS Proceedings