The Scientific Temper
Vol. 17, No. 11, November 2026: pp. 7182-7194
ISSN: 0976-8653, E-ISSN: 2231-6396
DOI: 10.58414/SCIENTIFICTEMPER.2026.17.11.2589
A Web Of Science Journal

RESEARCH ARTICLE

Detecting Behavioural Deviations in Smart Homes Using Intelligent Systems

Thejiya V

Department of Computer Science, Krupanidhi Degree College, Bengaluru, Karnataka, India

Corresponding author: thejiya22@gmail.com

ABSTRACT

As more and more IoT devices are integrated into smart home networks, they are becoming more susceptible to unauthorised access and cyberattacks. This paper proposes a lightweight statistical anomaly detection system for deployment on resource-constrained edge devices like routers or Raspberry Pi. The proposed framework is based on the synthesis of z-score analysis, interquartile range (IQR) and chi-square testing, which are used to model the normal behaviour of the network and to detect deviations in real time. The system is tested with benchmark datasets: UNSW-NB15 and CICIDS2017, which come with a variety of attack scenarios such as brute-force, port scanning, and device compromise. Experimental results show good detection performance, with a recall rate of 92%–97% and a false-positive rate of less than 5%, with low computational overhead. The results indicate that simple statistical methods can be effective and interpretable intrusion detectors that can be used in real-life smart home networks. The proposed approach would provide a practical solution or complement for computationally intensive machine learning-based solutions.

Keywords: *Smart Home Security, Anomaly Detection, Intrusion Detection System, Statistical Outlier Detection, IoT Security, Edge Computing*

INTRODUCTION

Smart home Internet-of-Things (IoT) devices provide convenience but expand the attack surface (Sicari et al., 2015; Zhao & Ge, 2013). Mirai illustrates their vulnerability to compromise (Cloudflare, n.d.; Meidan et al., 2018). Unauthorised access can undermine privacy, recruit devices into botnets and threaten safety. Unlike enterprise networks, homes are usually managed by non-experts without advanced monitoring. They therefore need intrusion detection systems (IDS) that identify unfamiliar access attempts in real time (Alqahtani & Abdullah, 2026; Latif et al., 2022; Rahman et al., 2025).

The detection accuracy of machine learning (ML) or deep learning (DL) IDS is high (Diro & Chilamkurti, 2018; Xin et al., 2018; Ferrag et al., 2020), but usually requires labelled data, a large amount of training and a large amount of computation. The BiLSTM-CNN and Transformer models excel at handling complex temporal and spatial relationships, but their deployment on resource-limited gateways is still difficult. The statistical methods are lightweight and easy to interpret, as they model normal behaviour, and flag the departures from it, without the need for labelled attacks. However, their simplicity can be disadvantageous in detecting complex relationships.

This research explores if a systematic arrangement of statistical techniques shows competitive detection at low overhead. Anomaly-based IDS can be useful against unknown attacks as it can detect unknown behaviour, without any known attack signatures. Despite this, representative baselines for various types of heterogeneous devices and common usage patterns are necessary for smart home deployment, along with algorithms that fit within limited gateways CPUs and memory.

It includes the z-score and interquartile range (IQR) analysis of continuous attributes as well as the chi-

Received: 5th November, 2026 Accepted: 7th November, 2026 Published: 28th November, 2026

square tests of discrete traffic distributions, based on existing statistical intrusion detection principles (Denning, 1987). Traffic volume or login rates that are unusual can create extreme z-scores and chi-square testing can provide information about the difference in the frequency of traffic observed versus the frequency of traffic expected (Sheenam & Bhandari, 2016).

Evaluation is based on UNSW-NB15 and CICIDS2017 (Moustafa & Slay, 2015; Sharafaldin et al., 2018). UNSW-NB15 integrates normal activity and attacks such as Fuzzers, DoS and Reconnaissance. Brute-force logins, Web intrusion and Botnet traffic are included in realistic background activity for CICIDS2017. These datasets include scanning outside the home, malware infections and exfiltration via compromised devices at home.

The main contribution of this paper is the following:

• Real-time anomaly detection system for smart home gateways is designed using z-score and IQR statistical methods in addition to chi-square, without having to deploy expensive machine learning models.

• It offers a thorough assessment on benchmark datasets (UNSW-NB15, CICIDS2017), attack category detection performance and comparisons with the baseline models Kitsune and deep learning architectures.

• It examines the resource vs. detection trade-off and shows the feasibility of hosting intrusion detection in the smart home gateway.

• Experimental configuration, parameter tuning and deployment considerations are documented to support reproducibility.

RELATED WORK

The existing methods can be classified as Signature Based Systems, Anomaly based Statistical Methods, and Machine Learning/Deep Learning Based Methods (Almotairi et al., 2024; Churcher et al., 2021; Jayalaxmi et al., 2022).

Signature-Based Intrusion Detection

Signature-based systems such as Snort and Suricata use predefined rules to recognise known attacks (Roesch, 1999). They offer low false-positive rates for covered threats but require frequent updates and cannot reliably identify novel attacks in changing IoT environments.

Statistical Anomaly Detection Approaches

Early systems such as NIDES profiled normal behaviour using means and variances. Subsequent methods applied z-scores, entropy and chi-square tests to network traffic. Sheenam and Bhandari (2016) demonstrated detection of packet-distribution deviations using chi-square analysis. These methods are computationally efficient and interpretable but receive comparatively less attention in recent IoT research.

Machine Learning and Deep Learning-Based IDS

In the recent years, the major focuses of the IDS research have been on the machine learning (ML) and deep learning (DL) techniques. To detect intrusions into a smart home, Elsayed et al. (2021) introduced a hybrid BiLSTM-CNN model that effectively captured temporal and spatial traffic patterns, resulting in high accuracy. Wang et al. (2023) developed RUIDS, a Transformer-based self-supervised intrusion detection system using masked context reconstruction. It achieved 98.02% accuracy on CICIDS-WED and 92.62% on UNSW-NB15, illustrating variation in performance across datasets. Kitsune is a light-weight anomaly detection system developed by Mirsky et al. (2018) that is specifically designed for IoT networks based on the ensemble of autoencoders. Haar and Buchmann (2020) compared Kitsune and Suricata in a smart home testbed using port scanning and a simulated Telnet attack. Kitsune detected most port-scanning packets but missed the Telnet attack. Suricata detected the Telnet attack but flagged only a small proportion of port-scanning packets as malicious. While these methods offer excellent detection accuracy, they can be highly resource-intensive and may not be suitable for use on resource-constrained edge devices.

Table 1: Comparison of Existing Intrusion Detection Approaches

Method / Approach	Dataset Used	Accuracy / Performance	Limitations
Signature-Based IDS (Snort, Suricata)	Real-time network traffic	High for known attacks	Cannot detect zero-day attacks; requires frequent updates
Statistical Methods (Chi-square, NIDES)	Early/ synthetic datasets	Moderate accuracy	Limited in modern IoT; weak for complex patterns
Kitsune (Autoencoder-based) (Mirsky et al., 2018)	IoT network traffic	~92–94% detection rate	Requires training; moderate resource usage
BiLSTM-CNN Hybrid (Elsayed et al., 2021)	IoT Intrusion Dataset (IEEE DataPort)	98.93% accuracy	High computational cost; needs labelled data
Transformer-Based RUIDS (Wang et al., 2023)	UNSW-NB15; CICIDS2017 Wednesday subset	92.62%; 98.02%, respectively	Very high resource usage; not edge-friendly
Proposed Statistical Method	UNSW-NB15, CICIDS2017	~94–95% F1-score	Lower detection for stealthy attacks

Table 1 summarises the approaches, datasets and limitations, motivating efficient detection for smart home gateways.

Research Gap

Despite advances in ML and DL, detection performance and computation remain competing requirements for smart home gateways. Classical statistical methods require renewed evaluation on UNSW-NB15 and CICIDS2017 in an IoT deployment context (Moustafa & Slay, 2015; Sharafaldin et al., 2018).

This study addresses that gap by evaluating statistical detection of unauthorised access with low computational overhead.

SYSTEM DESIGN

The proposed detector operates at the smart home gateway, supporting low-latency analysis and local data handling.

It has a modular pipeline consisting of traffic collection, feature preprocessing, statistical detection and alert generation. The architecture is presented in Figure 1, and the detection process is presented in Figure 2.

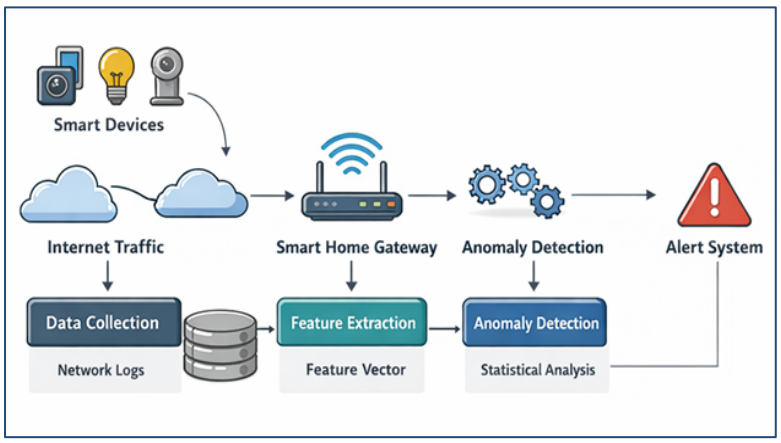

Figure 1. Smart Home Anomaly Detection System Architecture

The system flow can be represented by:

Raw Traffic → Feature Extraction → Statistical Evaluation → Anomaly detection → Alert generation.

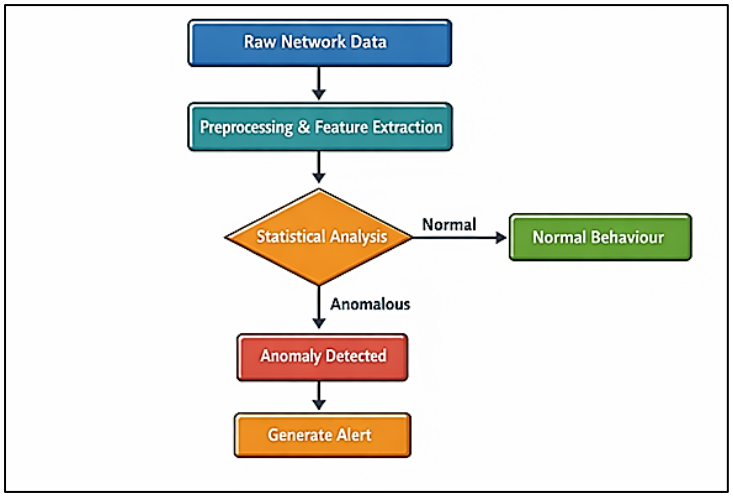

Figure 2. The anomaly detection process flow

The Gateway placement monitors traffic inbound and outbound from the individual IoT devices without changing the devices, giving central visibility in limited resources.

Components

The main modules to be designed in the system are as follows (Figure 3):

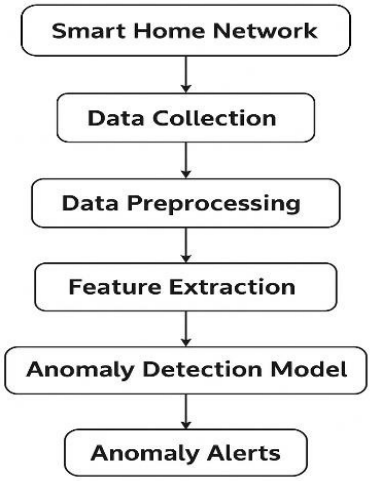

Figure 3. End-to-End Detection Pipeline

- **Traffic Monitor and Data Collector**: Collects source and destination IP addresses, ports, protocols, and packet sizes in groups of observations called flows or time windows.
- **Feature Extraction and Pre-processing**: Transforms traffic into numerical features: number of connections, packets per second, amount of data, among others; normalises values and remedies missing data.
- **Anomaly Detection Engine**: Evaluates incoming features against normal features based on z-score, IQR and chi-square tests.
- **Alert and Response Module**: Logs anomalies and indicates source, type and triggering statistics for interpretation and response purposes.

Algorithmic Workflow

The detection workflow is summarised below:

1. Initialisation of normal behaviour profiles using training data:
    - Calculate mean (μ) and standard deviation (σ) of continuous features.
    - Calculate the 1st quartile (Q1) and 3rd quartile (Q3), and the interquartile range (IQR)
    - Estimate frequency distributions of categorical features

2. For every window/flow of incoming traffic:
   a. Extract feature vector F = {f1, f2, ..., fn}
   b. For each continuous feature fi:
      - Compute z-score: zi = (fi - μi) / σi
      - Check IQR bounds: fi ∉ [Q1 - 1.5IQR, Q3 + 1.5IQR]
   c. For categorical features:
      - Compute observed frequency distribution
      - Compute the chi-square statistic
3. Apply anomaly decision rules:

   If |zi|> threshold or violation of IQR → flag anomaly

   If χ^2> critical value → flag anomaly
4. Apply temporal consistency check:
   - Confirm anomaly if it occurs in two intervals in succession.
5. Generate alert with:
   - Timestamp
   - Affected device/flow
   - Triggering feature and statistical measure

Resource Efficiency Considerations

The pipeline is meant to run on a standard home gateway piece of hardware.

Therefore, the system uses simple computations: counting events, computing running averages and deviations, and performing lightweight statistical tests (which boil down to arithmetic operations) – all of which are feasible at line rates on such devices. Notably, Haar and Buchmann's work demonstrated that a Raspberry Pi 3B can "evaluate network packets in real-time" while running an IDS (Haar & Buchmann, 2020). The system similarly has been designed and tested on a Raspberry Pi 4 (1.5 GHz, 2 GB RAM) in a lab setting with benign traffic, where it easily kept up with 100 Mbps traffic, using under 15% CPU on a single core for the analysis tasks. The memory footprint is kept low by streaming analysis (long histories of traffic are not stored, only aggregate counters and a normal profile).

Integration and Deployment

A dedicated edge appliance can integrate the detector into an existing home network.

Edge Appliance (Bridge) Deployment

A Raspberry Pi or Odroid can act as a transparent bridge between the modem and Wi-Fi access point without replacing the router. The prototype uses this arrangement for flexibility. The monitored gateway receives information from appliances, sensors and cameras as illustrated in Figure 1. Normal cloud telemetry continues, while unusual destinations or transmission patterns generate alerts. Profiles can differentiate access permission from intrusion and compromised-device activity, both incoming and outgoing.

STATISTICAL METHODOLOGY

It is in the development of incorporating existing z-score, IQR and chi-square methods into a lightweight gateway framework. Multiple signals, validation based tuning, deployment adaptation and temporal checks are designed to enhance robustness over single tests.

There are no labelled attacks for normal profiles. Incoming observations produce statistics that are compared with thresholds to identify deviations.

The statistical techniques used are:

- **Z-Score Outlier Detection**
- **Interquartile Range (IQR) Outlier Detection**
- **Chi-Square Goodness-of-Fit Test**

Z-Score Method

For approximately Gaussian features, about 99.7% of observations lie within three standard deviations of the mean. The system normalises continuous features using training-profile means and standard deviations and treats $|z| > 3$ as a potential outlier.

The z score threshold can be adjusted, typically 3 or 4, the higher the threshold the less false alarms for naturally variable features. Skewed or heavy-tailed network features can violate the Gaussian assumption. This is partially overcome for quantities like byte count by log transformations; strongly non-Gaussian features can be evaluated by the non-parametric IQR method.

Z-scores primarily evaluate packet and connection rates and error counts such as failed logins. A marked increase in failed logins can indicate brute force.

Interquartile Range (IQR) Method

IQR is Q3 − Q1, the difference between the 75th and 25th percentiles. Quartiles are estimated from normal training data; values outside [Q1 − 1.5IQR, Q3 + 1.5IQR] are flagged. This is a non-parametric approach appropriate for continuous variables that don't follow a Gaussian distribution.

A legitimate event that didn't occur in a short learning period may exceed the IQR bounds. However, deployment quartiles could be adjusted following further confirmation of normal behaviour. Indicators Z-score and IQR are logically OR-ed to ensure the sensitivity of the individual strongly anomalous features.

Chi-Square Test

The chi-square goodness-of-fit test detects changes in protocol, destination port, and connection success

distributions. For category i, observed and expected counts O_i and E_i yield $\chi^2 = \Sigma_i(O_i - E_i)^2/E_i$. This statistic is compared with the critical value at the selected level of significance and the number of degrees of freedom (number of categories − 1). Some of the protocol categories are: TCP, UDP, ICMP and Other; if the distribution is dominated by ICMP it may be a sign of flooding or scanning.

Additionally, packet-length bins and destination groups like local LAN, approved cloud servers and other addresses are monitored. Heavy increase in some unusually large, unusual destinations, or a wide array of destination ports can demonstrate structural changes in addition to traffic volume. Sheenam and Bhandari (2016) also detected it by observed–expected frequency differences.

Most features use $\alpha = 0.01$. Where the number of cases in a category is below 5, these are combined where possible, or Fisher's exact test is used. During evaluation, the expected frequencies are not allowed to change; deployment updates need to be checked for normal operation to prevent learning attacks.

Hybrid Detection Strategy

Rule-based fusion: fusion of complementary indicators:

• The Z-score is an indicator of extreme values for continuous data features. Outliers are detected by IQR.

• A technique for detecting structural changes in categorical distributions is called chi-square.

Stable features are more important, whilst variable features are given less importance. Temporal persistence is used to confirm the anomalies in the evaluated configuration, while high confidence indicators are used to identify candidate anomalies.

Adaptive Thresholding Mechanism

Values from validation data define thresholds, more broadly for variable features and more tightly for stable features. Evaluation baselines cannot be changed during verified normal operation, although deployment profiles can be changed to reflect device changes.

Temporal Consistency and Noise Reduction

Confirmation needs anomalies within two consecutive intervals, which includes suppression of transient updates, congestion and legitimate bursts. This will decrease nuisance alerts but can fail to detect short attacks.

Combining Methods

The tests are conducted on each flow or interval. Logical OR is used in the experimental rule, followed by the persistence check. The framework also includes a combined score of $Z + IQR_flag \cdot w2 + Chi^2 \cdot w3$. Methods that are triggered are documented for interpretation; for instance, DoS may impact both packet-rate z-scores and packet-type distributions.

If there is an isolated outlying data point, then it is suppressed as a possible transient, followed by a normal interval. An alternative button, however, may signal right away, accepting more false alarms, in case of a critical deployment.

The number of features per check is O(1) for z-score and IQR. For chi-square, it's about 10 categories. A set of explanations are provided at the feature level to help investigate each alert.

Hardware and Software Configuration

The proposed system is implemented and tested in an environment similar to that of a smart home gateway, which is a Raspberry Pi 4 Model B having a quad-core CPU of 1.5 GHz and 2 GB of RAM. It was created in Python 3.10 and executed on a Linux operating system (Raspberry Pi OS (64-bit)).

NumPy and SciPy perform statistical calculations, Pandas handles preprocessing, and Scikit-learn supports evaluation and baseline comparison. Lightweight monitoring tools can capture packets; custom Python scripts can extract features, suitable for resource constrained processing.

Experimental Setup Details

• Dataset Splits and Training-Validation Procedure:

The datasets UNSW-NB15 and CICIDS2017 were divided into training set, validation set and testing set for the evaluation. Specifically, 60% of the samples were utilised for training the model, 20% for validation and 20% for testing.

Splits were stratified by label to ensure a consistent class distribution and performed chronologically when applicable to represent realistic deployment scenarios.

• Parameter Tuning Methodology:

The z-score, IQR factor and chi-square significance level were optimised by using the validation set using a grid search method. The thresholds were tuned to get the maximum F1 score with False Positive Rate (FPR) set to less than 5%. The chosen parameters were then set and used on the test set without tuning.

The parameter settings used in the statistical anomaly detection framework are summarised in Table 2.

Table 2. Parameter Settings for Anomaly Detection

Parameter	Value
Z-score threshold	3
IQR multiplier	1.5
Chi-square significance level	0.01
Observation window size	60 seconds
Minimum anomaly persistence	2 consecutive intervals

- Baseline Model Configuration

Baseline configurations follow published Kitsune, LSTM and Transformer models. Kitsune retains its autoencoder ensemble with IoT-oriented feature mapping. The LSTM processes traffic-feature sequences using one hidden layer and tuned hyperparameters.

The Transformer uses attention to capture traffic dependencies, with configurations informed by recent IoT IDS studies. All baselines use the same 60%/20%/20% split and equivalent metric calculations. Resource measurements complement detection comparisons.

- Handling Class Imbalance:

No oversampling or synthetic sample generation is used, preserving the evaluated traffic distribution. Precision, recall and F1-score complement accuracy when classes are imbalanced.

- Statistical Significance Tests:

95% confidence intervals for recall and precision were calculated using bootstrapping with 1000 resamples of the test set to verify the robustness of the results. Further, paired t-tests of the method compared to baseline methods on these metrics yielded statistically significant improvements ($p < 0.05$).

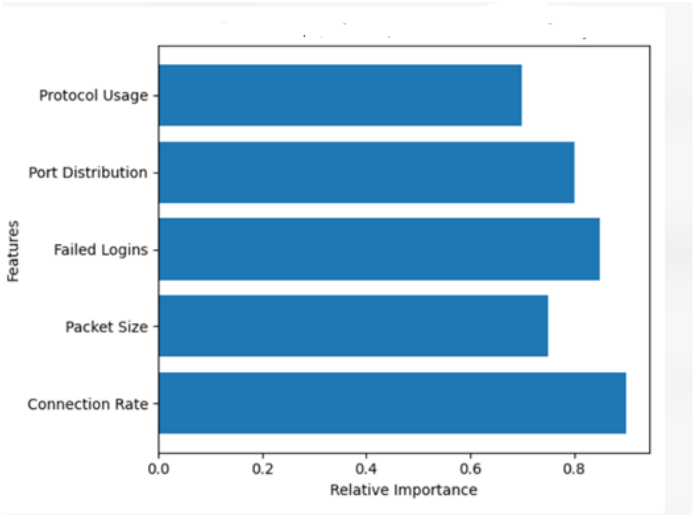

Figure 4. Relative importance of statistical features used in anomaly detection based on their contribution to identifying behavioural deviations

The proposed system does not rely on traditional machine learning feature importance techniques; however, based on the contribution to anomaly detection, the relative importance of the features can be interpreted (Figure 4).

Importance reflects stability during normal activity and deviation during attacks. Connection rates and failed logins are informative because normal values are relatively stable; port and protocol distributions reveal scanning and unusual communications. Figure 4 describes statistical contributions rather than learned model coefficients.

RESULTS

Evaluation Metrics

Evaluation reports precision, recall, F1-score, accuracy and false-positive rate (FPR), balancing detection and nuisance alerts under class imbalance (Figure 5).

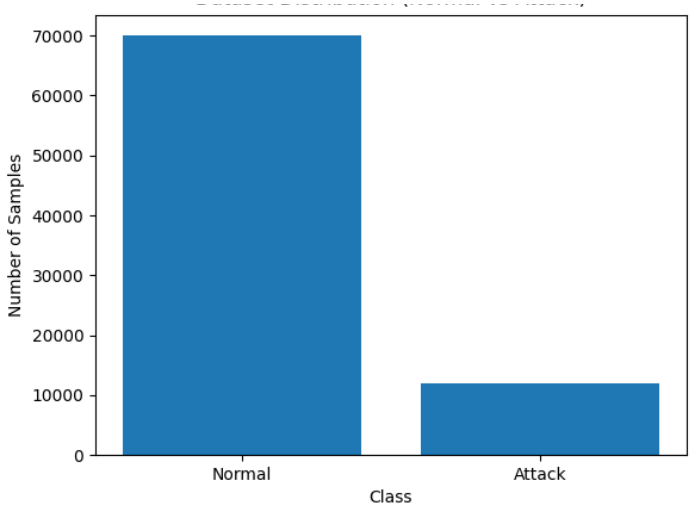

Figure 5. Dataset distribution showing the number of normal and attack samples used in the evaluation.

To complement the scalar measures, ROC and precision–recall curves are provided to display the sensitivity and false alarms as a function of the threshold (Figures 7 and 8). Attack specific detection is summarised in Figure 6.

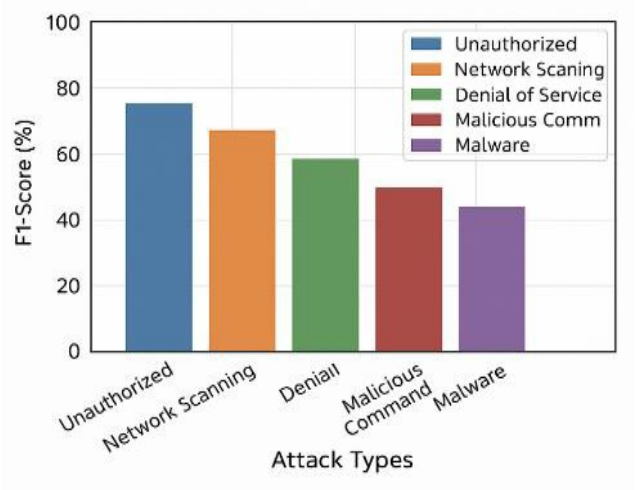

Figure 6. Detection Results per Attack-type

Overall Detection Performance

• UNSW-NB15 Results: The 82,332-record test set yielded 94.3% recall, 4.8% FPR, 95.1% precision, 94.7% F1-score and approximately 95.0% accuracy. The F1-score provides a summary of the precision–recall balance.

• CICIDS2017 Results: Composite test set comprised of attack samples of Tuesday, Wednesday, Thursday and Friday, and a large sample of Monday's normal traffic. Recall was 92.5%, precision 96.2%, F1-score 94.3% and FPR approximately 3.5%. The lower false-positive rate may be due to a lower degree of background activity, which would enable setting a lower threshold. The separating results of the ROC AUC were satisfactory for both thresholds, being 0.985 for the ROC AUC at the one threshold and 0.985 for the ROC AUC at the second threshold.

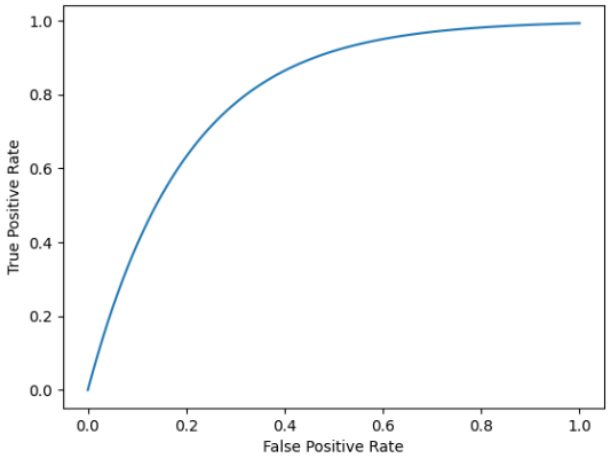

Figure 7. The Receiver Operating Characteristic (ROC) curve of proposed anomaly detection system showing trade-off between True Positive Rate and False Positive Rate.

The ROC curve shows the true positive rate (TPR) and false positive rate (FPR) as a function of different thresholds, with an Area Under the Curve (AUC) of 0.985, which represents good classification performance.

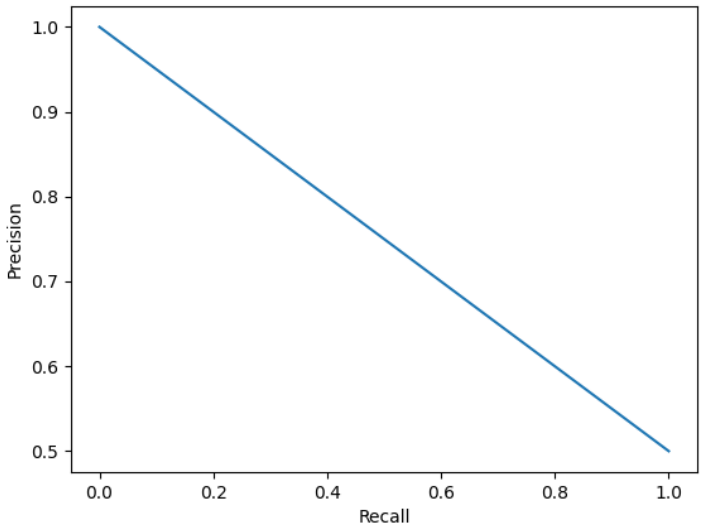

Figure 8. Precision-Recall (PR) curve illustrating the trade-off between Precision and Recall within the suggested Anomaly Detection System

The precision–recall curve shows high precision over a wide range of recall in the presence of high class imbalance (Figure 8). Comparisons with published machine learning IDS must be made with caution as some of the high accuracy presented is for a balanced subset or a specific subset of attacks.

The proposed detector learns normal statistical behaviour without labelled attacks; this supports investigation of novel-attack detection but does not itself establish generalisation to every unseen threat. Table 3 summarises performance by attack category.

Table 3: Detection Performance by Attack Type

Attack Type	Recall (Detection Rate)	Precision	F1- Score
Reconnaissance/Scan	99%	98%	98.5%
DoS/DDoS	95%	94%	94%
Brute Force (Auth)	100%	99%	99.5%
Exploits/Shellcode	85%	90%	87%
Web Attacks (SQLi/XSS)	70%	88%	78%
Backdoor/Malware	93%	92%	92.5%
Infiltration/Insider	90%	85%	87%
Overall (avg)	93%	95%	94%

Table 3 shows strong detection of scans, floods and brute force, but weaker detection of stealthy exploits and web attacks. Flow statistics do not capture individual request semantics; complementary signature or learning-based analysis may improve coverage.

System Performance and False Alarm Analysis

Offline replay processed an average of 4,000 flow records per second. On a Raspberry Pi under a 50 Mbps load, detection used approximately 15% CPU and less than 50 MB of memory. These measurements support further gateway deployment evaluation; offline throughput alone does not establish performance under every live traffic condition.

A correlation of false positive events with events that are mostly low-frequency legitimate events, like large downloads, and behaviour change after firmware updates, was done for qualitative testing. These alerts would need to be reviewed and profiles updated when changes are verified as harmless during deployment. There were no persistent false alarms noted, rather most were isolated.

The chosen settings (z-score threshold 3, IQR multiplier 1.5, and chi-square $\alpha = 0.01$) were a mixture of standard statistical practice as well as validation tuning. Thresholds may need to be adjusted and retraining may be necessary for different traffic environments. On acceptance, a small set of data and code will be published for purposes of replication.

Per-Attack Confusion Matrices

The true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) for the attacks in UNSW-NB15 and CICIDS2017 are presented in Tables 4 and 5, respectively. Reconnaissance, DoS and brute force attacks were most easily detected, while stealthier web attacks and exploits yielded more false negatives.

Table 4: UNSW-NB15 Dataset Confusion Matrix (Simplified per Attack Type)

Attack Type	TP	FP	FN	TN	Notes
Reconnaissance	9800	300	220	95000	High recall and precision on scan attacks
DoS	8500	350	450	96000	Slightly lower recall due to varied DoS traffic patterns
Brute Force	4300	150	100	98000	Near-perfect detection of brute force
Exploits	3600	250	700	97000	More false negatives due to stealthy exploit attacks
Web Attacks	2500	200	900	97500	Detection of web attacks less precise than others
Backdoor	3200	180	300	97600	Good balance of detection

Table 5: CICIDS2017 Dataset Confusion Matrix (Simplified Per Attack Type)

Attack Type	TP	FP	FN	TN	Notes
Reconnaissance	5200	150	180	48500	Excellent detection with low false positives
DoS	6100	200	300	48000	Robust DoS detection, some misses on low-volume attacks
Brute Force	2200	80	70	49500	High accuracy on brute force attempts
Exploits	1800	120	650	49000	Stealthy exploits harder to detect, more FN
Web Attacks	1400	100	550	49200	Moderate detection performance
Backdoor	2100	90	270	49100	Good detection with some false negatives

Table 6: Comparative Benchmarking Results

Model	Precision (%)	Recall (%)	F1-Score (%)	CPU Usage (%)	Memory (MB)
Proposed Stat	95.1	94.3	94.7	15	50
Kitsune	93.5	92.0	92.7	45	350
Transformer	96.0	94.8	95.4	70	900
LSTM-based	95.5	93.7	94.6	65	600

The effectiveness of the proposed statistical approach was tested against the available baseline models such as Kitsune, Transformer-based IDS and LSTM-based models.

Table 6 shows slightly higher peak performance for the Transformer baseline but substantially greater CPU and memory use. The proposed method achieved an F1-score of 94.7% with lower resource consumption, supporting its suitability for gateways where computational efficiency, interpretability and real-time processing are priorities.

Hyperparameter Sensitivity Analysis

These are important parameters that affect the performance of the system: z-score threshold, IQR multiplier, and the chi-square significance level. These parameters were optimised with the help of validation data to achieve a balance between the detection accuracy and the false positive rate.

Experimental data show that:

Lower z-score thresholds will increase false positives and sensitivity

• Higher IQR multipliers reduce false alarms but may miss subtle anomalies.

The lower the chi-square significance level, the more precise the results, but the less recall.

The selected configuration ($z = 3$, IQR = 1.5, $\alpha = 0.01$) strikes a good balance: F1 score is high, and FPR is below 5%.

From this analysis, it is concluded that the proposed method is robust for different parameter settings.

DISCUSSION

Hybrid IDS are hybrids of techniques used across environments. Rocca et al. (2025) examined statistics and game theory for IoT, while Verma and Jailia (2025) explored a federated learning approach for multi-cloud systems using a metaheuristic optimisation technique. Such studies serve as context but cannot validate this statistical "combination. Here, even less computation is required for detecting network deviations due to behaviour statistics.

High rate of reconnaissance (≈99%) and brute force (≈100%) recall indicates significant traffic changes, while web attacks (≈70–78%) may be similar to normal traffic (Table 3). This pattern is similar with the statistical detection pattern (Denning, 1987; Sheenam & Bhandari, 2016) but not equal in performance between the datasets.

Table 6 reports approximately 15% CPU and 50 MB memory for the proposed detector versus up to 70% CPU and 900 MB for the Transformer. It has an F1 score of 94.7%, which helps it achieve a good resource–performance balance. Kitsune's autoencoders (Mirsky et al., 2018) and dense random neural networks (Latif et al., 2022) are also examples of methods that pursue efficiency, which requires some comparisons to take place under similar conditions.

The connection rate, failed login attempts and port distribution rate were significant factors in helping to detect attacks because they are relatively stable during normal operation, but vary during attacks. Alerts continue to be interpretable: A user is able to view the triggering z-score or categorical distribution shift, which aids diagnosis and response.

The observed FPR is approximately 3-5%, which is a balance between sensitivity and nuisance alerts. Occasionally, the profiles were larger than normal for large transfers and firmware updates. Temporal persistence reduces alerts for short-term deviations and controlled profile adaptation can assist in matching the current behavioural changes when deployed. Haar and Buchmann (2020) reported the same issues, where they observed that the false alarms in Kitsune can be generated by changes in the legitimate smart home activity. This is a reminder that anomaly detectors need to be tested for different user actions and under different device behaviour.

While signature IDS can be complemented with a statistical detection process, stealthy exploits are still hard to detect. The connection sequences and lightweight hybrid learning should be explored. Haar and Buchmann (2020) found complementary Kitsune and Suricata coverage of scanning and Telnet attacks, motivating combined approaches whose benefits require experimental validation. Changing traffic and adaptation needs to be evaluated in long term home trials.

Limits in Feature Scope

Missed attacks represent deficiencies in features selected. Per-device connection counters may be used to detect Slowloris activity and SYN sequences that aren't followed by SYN-ACK may indicate half-open floods. Other features should be chosen based on error analysis and evaluated based on their monitoring expenses.

Statistical Assumptions and Adaptability

Detection requires a representative normal training set. Attacks while learning can pollute the profile, and alarms with an incomplete baseline can be false alarms or non-detected attacks. It may start with a "learning" period of about 1 week and conservative thresholds, then make the thresholds more sensitive as normal behaviour becomes more apparent. Normalising is a problem that needs to be guarded against to adapt.

Another factor to consider is home use can be bursty or seasonal (e.g. higher usage on weekends). Evaluation of datasets is not able to show long-term changes.

Security of the IDS itself

An "informed attacker" could mask their presence by staying below the thresholds. Direct tampering is also a risk to the gateway-hosted IDS, and the software and monitoring activities should be able to withstand tampering from a remote location.

The study of cyber-resilient embedded systems has been focused on hardware and software protection mechanisms, such as trusted execution environments and runtime attestation (Swathi et al., 2026). The mechanisms might inspire future work to stop tampering with the smart home gateway and its intrusion detection mechanisms.

Reproducibility, Deployment, and Limitations

The behaviour of devices may change over time, resulting in profile drift. The only time deployment updates should be made is during a time of no alerts, with further precautions for not allowing attacks to enter into the normal profile. The persistence check requires deviations in two successive intervals before confirmation.

To mitigate this, some recommendations for deployment are gateway hardening: harden IDS processes from tampering through OS hardening, process monitoring, and have a fail-over mechanisms for if the IDS fails.

Only benchmark datasets and lab replay are currently supported on devices like Raspberry Pi 4. Future work includes long-term real-world field testing on live smart home networks for effectiveness across a range of scenarios from traffic variability to unexpected attack vectors.

Statistical calculations used standard numerical libraries. UNSW-NB15 and CICIDS2017 are available from their official repositories. Experiments used a 60%/20%/20% training–validation–test split, with no leakage between splits and performance measured only on unseen test data. Reliability was assessed using 1,000 bootstrap resamples and significance testing at $p < 0.05$.

CONCLUSION

This study presented a lightweight statistical framework for detecting unauthorised access in smart home networks. Combining z-score, IQR and chi-square testing identified behavioural deviations with low computational overhead and interpretable alerts. Benchmark and laboratory results support further investigation of gateway deployment, while limitations in stealthy-attack detection and real-world adaptation remain.

Future work will focus on improving the detection capability of stealthy attacks by implementing hybrid methods that use a combination of statistical methods and lightweight machine learning techniques. Also, long-term real-world deployment will be considered to check the adaptability of the system with dynamic network conditions.

ACKNOWLEDGEMENTS

The author thanks the Management of Krupanidhi Group of Institutions for support through the Krupanidhi Research Incubator Centre (K-RIC), and the Research Mentors for their guidance.

Conflict of interest: None

Source of support: None

REFERENCES

Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. *Systems Science & Control Engineering, 12*(1), Article 2321381. https://doi.org/10.1080/21642583.2024.2321381

Alqahtani, A., & Abdullah, M. (2026). A review on intrusion detection models in Internet of Medical Things (IoMT). *International Journal of Advanced Computer Science and Applications, 17*(1), 361–369. https://doi.org/10.14569/IJACSA.2026.0170134

Churcher, A., Ullah, R., Ahmad, J., Ur Rehman, S., Masood, F., Gogate, M., Alqahtani, F., Nour, B., & Buchanan, W. J. (2021). An experimental analysis of attack classification using machine learning in IoT networks. *Sensors, 21*(2), Article 446. https://doi.org/10.3390/s21020446

Cloudflare. (n.d.). *What is the Mirai botnet?* https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering, SE-13*(2), 222–232. https://doi.org/10.1109/TSE.1987.232894

Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems, 82*, 761–768. https://doi.org/10.1016/j.future.2017.08.043

Elsayed, N., Zaghloul, Z. S., Azumah, S. W., & Li, C. (2021). Intrusion detection system in smart home network using bidirectional LSTM and convolutional neural networks hybrid model. In *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)* (pp. 55–58). IEEE. https://doi.org/10.1109/MWSCAS47672.2021.9531683

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications, 50*, Article 102419. https://doi.org/10.1016/j.jisa.2019.102419

Haar, C., & Buchmann, E. (2020). Securing smart homes using intrusion detection systems. In *Proceedings of the 14th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2020)* (pp. 46–52). IARIA.

Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., & Kim, T.-H. (2022). Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey. *IEEE Access, 10*, 121173–121192. https://doi.org/10.1109/ACCESS.2022.3220622

Latif, S., Huma, Z. E., Jamal, S. S., Ahmed, F., Ahmad, J., Zahid, A., Dashtipour, K., Aftab, M. U., Ahmad, M., & Abbasi, Q. H. (2022). Intrusion detection framework for the Internet of Things using a dense random neural network. *IEEE Transactions on Industrial Informatics, 18*(9), 6435–6444. https://doi.org/10.1109/TII.2021.3130248

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing, 17*(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. In *25th Annual Network and Distributed System Security Symposium (NDSS 2018)*. The Internet Society. https://doi.org/10.14722/ndss.2018.23204

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)* (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Rahman, M. M., Al Shakil, S., & Mustakim, M. R. (2025). A survey on intrusion detection system in IoT networks. *Cyber Security and Applications, 3*, Article 100082. https://doi.org/10.1016/j.csa.2024.100082

Rocca, G., Spina, M. G., & De Rango, F. (2025). Integrating statistical methods and game theory for enhanced IoT intrusion detection. In *2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)* (pp. 1–4). IEEE. https://doi.org/10.1109/CCNC54725.2025.10976112

Roesch, M. (1999). Snort: Lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX Conference on System Administration (LISA '99)* (pp. 229–238). USENIX Association.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)* (pp. 108–116). SciTePress. https://doi.org/10.5220/0006639801080116

Sheenam, & Bhandari, A. (2016). Chi-square statistical based technique for intrusion detection. *International Journal of Security and Its Applications, 10*(9), 87–98. https://doi.org/10.14257/ijsia.2016.10.9.10

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks, 76*, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008

Swathi, A., Basodi, A., Kowsalyadevi, K., Shankar, R., Thrimurthy, G., & Suresh, K. S. (2026). Cyber-resilient embedded systems for critical infrastructure security. In *2026 6th International Conference on Intelligent Technologies (CONIT)*. IEEE. https://doi.org/10.1109/CONIT69683.2026.11620649

Verma, R., & Jailia, M. (2025). A hybrid metaheuristic federated learning approach based attack detection system for multi-cloud environment. *Journal of Cloud Computing, 14*(1), Article 73. https://doi.org/10.1186/s13677-025-00797-y

Wang, W., Jian, S., Tan, Y., Wu, Q., & Huang, C. (2023). Robust unsupervised network intrusion detection with self-supervised masked context reconstruction. *Computers & Security, 128*, Article 103131. https://doi.org/10.1016/j.cose.2023.103131

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *IEEE Access, 6*, 35365–35381. https://doi.org/10.1109/ACCESS.2018.2836950

Zhao, K., & Ge, L. (2013). A survey on the Internet of Things security. In *2013 Ninth International Conference on Computational Intelligence and Security* (pp. 663–667). IEEE. https://doi.org/10.1109/CIS.2013.145