

Continuous Security Assurance Framework for Smart Homes: Mitigating SysNet Blindspot through STRIDE-Aligned Intrusion Detection and Robust Security Posture Quantification

By

Shahbaz Ali Imran

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy (PhD)

Department of Computer Science

School of Architecture, Built Environment, Computing and Engineering

Birmingham City University

Birmingham, United Kingdom

March 2026

Supervisor(s): Prof. Junaid Arshad, Dr. Yussuf Ahmed,

Dr. Fuad A. Ghaleb

COPYRIGHT STATEMENT

- The author of this thesis (including any appendices and/or supplementary materials) retains copyright ownership. However, the author grants Birmingham City University the right to use this work for administrative, promotional, educational, and teaching purposes.

- Copies of this thesis, in whole or in part, may only be made in accordance with the regulations of the University Library. Further details of these regulations can be obtained from the Librarian. This statement must be included in any copies produced.

- The ownership of any patents, designs, trademarks, or other intellectual property rights (excluding copyright), as well as any reproduced materials such as figures, graphs, or tables, may belong to third parties. Such materials must not be used or reproduced without prior written permission from the respective rights holders.

DECLARATION

I declare that this thesis is my own work and has not been submitted for the award of any other degree or qualification.

Where the work of others has been used, it has been properly acknowledged and referenced. Any material taken from published or unpublished sources has been clearly identified in the thesis.

I used generative AI tools only for limited support with formatting, LaTeX troubleshooting, grammar, and clarity of expression. No AI tool was used to generate the research findings, analysis, experimental results, data, references, or conclusions. The thesis content, interpretation, and final submitted work remain my own.

One paper related to this research has been accepted or published in the IRICT conference:

> Shahbaz Ali Imran, Junaid Arshad, Yussuf Ahmed, and Fuad A. Ghaleb, "From Data to Decision: A Framework for Continuous Security Assurance and Live Security Posture Management in IoT," IRICT Conference, [2025].

Signature: Shahbaz Ali Imran

Date: 08/May/2026

ABSTRACT

Smart homes have become high-frequency targets for cyber threats, yet the security mechanisms designed to protect them remain structurally incomplete. Existing intrusion detection systems are predominantly network-centric, monitoring traffic flows while remaining blind to threats that manifest exclusively at the device level through abnormal process execution, resource abuse, or file manipulation. This architectural gap, defined in this research as the **Sys-Net Blind Spot**, prevents any single-plane monitoring strategy from achieving comprehensive, continuous visibility over a smart home's true security condition.

This thesis addresses that gap through the design, implementation, and empirical validation of a **Continuous Security Assurance (CSA)** framework for IoT-enabled smart homes. The framework integrates synchronised system-level and network-level telemetry through a four-layer pipeline: a statistical hybrid profiling layer that establishes per-device behavioural baselines, a profiling-grounded anomaly detection layer that quantifies deviations as a composite score, a STRIDE-aligned Random Forest intrusion detection layer that maps validated anomalies to structured threat categories, and a physical consistency verification layer that maintains detection integrity under adversarial evasion. Together, these layers produce a continuously updated, quantifiable security posture score for each device and for the smart home as a whole.

A principal research contribution is **SmartHome-StrideDS**, a purpose-built, time-synchronised dataset integrating host-level and network-level telemetry under four STRIDE-aligned attack scenarios, Denial of Service, Spoofing, Tampering, and Information Disclosure, each designed to suppress observable signals in at least one data plane,

thereby directly targeting the detection challenge posed by the Sys-Net Blind Spot.

Empirical evaluation on SmartHome-StrideDS demonstrates that the Random Forest classifier achieves an overall F1-score of 0.96 and a classification accuracy of 96.42%, with per-class F1-scores of 0.96 for DoS, 0.89 for Spoofing, 0.89 for Tampering, and 0.97 for Information Disclosure. Random Forest exceeds the next-best evaluated classifier, MLP at 79.00%, by 17.42 percentage points in classification accuracy. Modality ablation further shows that the hybrid Sys-Net configuration achieves higher per-class F1-scores than both network-only and host-only configurations across all four evaluated attack categories.

Adversarial robustness is evaluated under both oblivious and adaptive black-box attacker models across three independent datasets. The CSA consistency verification layer reduces the net attack success rate from 66.6% (classifier-only) to 22.2% under a standard Random Forest, and from 16.5% to 6.4% under adversarial training, at a maximum perturbation budget of $\varepsilon = 0.15$. On the TON-IoT network corpus, near-complete interception is achieved (CSA-ASR = 0.0% at $\varepsilon = 0.15$), supporting cross-dataset transfer of the physical consistency verification architecture with corpus-appropriate rule calibration.

This research makes four contributions to knowledge: a formalised CSA architecture that elevates hybrid profiling from an implicit preprocessing step to a first-class analytical layer; the SmartHome-StrideDS dataset as a reproducible community resource for cross-layer IoT security research; empirical evidence that synchronised Sys-Net fusion improves detection performance across the four evaluated STRIDE attack categories relative to the corresponding single-modality configurations; and a cross-dataset evaluated, physically constrained adversarial defence principle assessed under both oblivious and adaptive threat models. Collectively, these contributions establish a practical and scalable foundation for real-time, interpretable, and adversarially robust continuous security assurance in IoT-enabled smart homes.

Keywords: Internet of Things (IoT), Smart Home Security, Continuous Security As-

surance, Anomaly Detection, Hybrid Threat Detection, Real-time Monitoring, Security Posture Quantification, STRIDE Threat Model, Adversarial Robustness, Black-Box Attack, Isolation Forest.

DEDICATION

This work is dedicated to my parents, brothers and teachers, whose unwavering support and guidance have been a constant source of strength throughout every stage of my life.

Challenges are an inevitable part of life; however, it is through learning, resilience, and perseverance that individuals grow stronger. The ability to face, manage, and overcome these challenges is often made possible by the support of those who stand beside us.

Their encouragement, wisdom, and moral guidance have played a fundamental role in shaping this journey. Without such support, even the greatest ambitions can seem unattainable. It is through these guiding principles and the strength they inspire that one is able to overcome even the most formidable obstacles.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to all those who have supported and contributed to the successful completion of this research. Their guidance, encouragement, and unwavering support have been instrumental in shaping my academic journey and transforming this work into a reality.

First and foremost, I would like to extend my heartfelt appreciation to Birmingham City University for providing me with the opportunity to pursue my PhD and for offering an environment rich in academic resources, knowledge, and intellectual growth. The University's commitment to excellence and its supportive research culture have played a vital role in my development.

I am deeply indebted to my supervisors, Prof. Junaid Arshad, Dr. Yussuf Ahmed, Dr. Fuad A. Ghaleb, and Dr. Mark Joseph, for their invaluable guidance, continuous encouragement, and expert insight throughout this research journey. Their constructive feedback, patience, and dedication have been essential in refining my ideas and strengthening the quality of this work. I am truly grateful for their mentorship, support, and the confidence they have placed in me. Their professionalism and commitment to research have been a constant source of inspiration.

Finally, I would like to express my sincere thanks to everyone who has contributed, directly or indirectly, to this work. Your support, encouragement, and belief in my abilities have been deeply appreciated, and I feel privileged to have been supported by such remarkable individuals throughout this journey.

Contents

Declaration **ii**

Page

1 Introduction **1**

1.1 Background and Motivation 1

1.1.1 Motivation for a Unified CSA Framework 6

1.2 The Need for Continuous Visibility in Smart-Home Security 6

1.3 The Sys-Net Blind Spot in Smart-Home Continuous Security Assurance . 8

1.4 Problem Statement 10

1.5 Research Aim, Objectives, and Research Questions 12

1.6 Scope 14

1.7 Thesis Contributions 15

1.8 Thesis Outline 18

1.9 Chapter Summary 19

2 Literature Review **20**

2.1 Introduction 20

2.1.1 Review Questions 23

2.1.2 Information Sources and Search Strategy 23

2.1.3 Eligibility Criteria 25

2.1.4 Full-Text Screening, Quality Assessment and Data Extraction . . 28

2.2 Security Assurance in Smart-Home Environments 29

2.2.1 Smart-Home Architecture and Threat Landscape 29

2.3 Profiling in Security Assurance 30

2.3.1 Profiling in Traditional Security Assurance 31
2.3.2 Profiling in Continuous Security Assurance 35
2.3.3 Comprehensive Analysis of Hybrid System–Network Behavioural Profiles 40
2.4 Anomaly Detection in Security Assurance 42
2.4.1 Anomaly Detection in Traditional Security Assurance 43
2.4.2 Anomaly Detection in Continuous Security Assurance 45
2.4.3 Comprehensive Analysis of Anomaly Detection Models 49
2.4.4 Comparative Analysis of Black-Box Machine Learning Techniques in IoT Anomaly Detection 51
2.5 Intrusion Detection Systems (IDS) in Security Assurance 56
2.5.1 IDS in Traditional Security Assurance 56
2.5.2 IDS in Continuous Security Assurance 57
2.5.3 Comprehensive Analysis of IDS and STRIDE Approaches 60
2.6 Synthesis: Limitations of Traditional Security Assurance 64
2.7 Rationale for Continuous Security Assurance (CSA) 64
2.8 Adversarial Robustness in IoT Intrusion Detection 65
2.9 Key Research Challenges 67
2.10 Chapter 2 Summary and Identified Research Gap 68

3 Methodology **70**
3.1 Introduction 70
3.2 Problem Formulation and Solution Concept 71
3.2.1 The Sys-Net Blind Spot in Smart-Home Security 71
3.2.2 Architecture of the Continuous Security Assurance Framework . . 72
3.3 Research Framework 73
3.3.1 Phase 1 — Hybrid Profile Construction 73
3.3.2 Phase 2 — Profiling-Grounded Anomaly Detection 75
3.3.3 Phase 3 — Anomaly-Aware IDS Model 77
3.3.4 Phase 4 — Continuous Security Assurance 78

3.4 Experimental Setup 80
3.4.1 Hardware and Software Configuration 80
3.4.2 Network Architecture 81
3.4.3 CSA Consistency Verification Layer: Operational Overview 82
3.5 Methodological Design Rationale and Novelty 83
3.5.1 Architectural Requirements and Design Alternatives 83
3.5.2 STRIDE Alignment and Threat Interpretation 84
3.5.3 Intrusion Detection Model Selection 85
3.5.4 Architectural Novelty and Methodological Boundaries 86
3.6 Evaluation Design and Procedures 87
3.6.1 Datasets, Feature Preparation, and Experimental Partitions . . . 87
3.6.2 Supervised Classification and Anomaly-Detection Protocols 88
3.6.3 Evaluation Metrics and Comparison Criteria 89
3.6.4 Modality Ablation and Baseline Classifier Comparisons 89
3.6.5 Adversarial Threat Models and Perturbation Procedures 89
3.6.6 Cross-Dataset Evaluation and Validity Boundaries 90
3.7 Chapter Summary 91

4 Design and Implementation of the Continuous Security Assurance Framework 92

Introduction 92
4.1 Hybrid Behavioural Profiling 93
4.2 The SmartHome-StrideDS Dataset 94
4.2.1 Data Collection 94
4.2.2 Attack Simulation and Data Labelling 95
4.2.3 Feature Selection Methodology and Exclusion Rationale 98
4.3 Feature Engineering 99
4.3.1 System-Level Features 99
4.3.2 Network-Level Features 100
4.3.3 Security Relevance of the Joint Feature Set 101

4.4 Statistical Behavioural Profiling Model 103
4.5 Profiling-Grounded Anomaly Detection 105
4.6 Behavioural Deviation Detection Framework 106
4.7 Anomaly Detection Model 107
4.7.1 Statistical Foundation 107
4.7.2 Composite Anomaly Score 108
4.7.3 Threshold and Temporal Filtering 109
4.8 Machine Learning Models for Anomaly Detection 110
4.8.1 Unsupervised Anomaly Detection Algorithms 111
4.9 STRIDE-Aligned Intrusion Detection 112
4.10 IDS Architecture 113
4.11 STRIDE Threat Categorisation 115
4.11.1 Formal Specification 115
4.11.2 Feature–Intent Relationships 117
4.12 Continuous Security Assurance Integration 118
4.13 CSA Framework Integration 119
4.14 CSA Consistency Verification Layer 121
4.14.1 Rule 1: Temporal Statistical Consistency 121
4.14.2 Rule 2: Physical Causal Consistency 122
4.14.3 Rule 3: STRIDE Semantic Viability 123
4.14.4 Isolation Forest Confirmation Gates 124
4.14.5 The Master Decision Gate 125
4.15 Security Posture Quantification Model 127
4.16 Real-Time Monitoring Dashboard 129
4.17 End-to-End Pipeline Operation 130
4.17.1 Integrated Pipeline Operation 130
Chapter Summary 132

5 Experimental Results and Validation **133**

5.1 Introduction 133

5.2 Dataset Characteristics and Profiling Validation 134
5.2.1 Validation via Baseline Anomaly Detection 134
5.3 Anomaly Detection Results 135
5.3.1 Hybrid Sys-Net Dataset 135
5.3.2 TON Linux Process Dataset 136
5.4 IDS Classification Results 137
5.4.1 Comparative Performance of Classifiers 137
5.5 Existing Comparison Evidence 140
5.6 Scenario-Based CSA Evaluation 142
5.6.1 Normal Operation 143
5.6.2 Standard STRIDE Attack 144
5.6.3 Adaptive Adversarial Attack 144
5.7 Validating the Sys-Net Approach: A Quantitative Comparison of Data Modalities 146
5.8 Classifier Results and Model Selection 149
5.8.1 Baseline Methods and the Proposed Framework 149
5.8.2 Fragility of Purely Statistical Machine Learning Under Adversarial Pressure 151
5.8.3 Preservation of Network Causality Under Feature Manipulation . 152
5.8.4 Dimensional Dead Space and Isolation Forest Gates 152
5.9 Adversarial Robustness Evaluation: Forensic Breakdown 153
5.10 Adversarial Robustness Evaluation 159
5.10.1 Classifier Fragility and the Contribution of Adversarial Training . 160
5.10.2 Cross-Dataset Validation: Generalisation to TON-IoT Host and Network Corpora 161
5.10.3 Gate Activation Analysis Across Datasets 164
5.10.4 Consolidated Summary: ML-ASR versus CSA-ASR Across All Configurations 165
5.10.5 Comparison with Existing Adversarial-Robustness Studies 166

5.10.6 Pipeline Latency . . . 171
5.11 Chapter Summary . . . 171

6 Discussion, Conclusions and Future Work 172

6.1 Synthesis of Findings . . . 172
6.2 Interpretation of Profiling Results . . . 173
6.3 Interpretation of Anomaly Detection Results . . . 174
6.4 Interpretation of IDS Results . . . 176
6.4.1 Implications . . . 177
6.5 Interpretation of Key Findings with Respect to Research Objectives . . . 177
6.5.1 Hybrid Sys-Net Profiling for Behavioural Baselines . . . 177
6.5.2 SmartHome-StrideDS as an Empirical Resource . . . 178
6.5.3 Profiling-Grounded Anomaly Detection . . . 179
6.5.4 STRIDE-Aligned Intrusion Detection . . . 179
6.5.5 Continuous Security Assurance Evaluation . . . 180
6.6 Profiling Contribution: What the Results Actually Show . . . 181
6.7 Anomaly Detection: A Comparison with Existing Approaches . . . 181
6.8 IDS Contribution: Why Architecture Outweighs Classifier Choice . . . 182
6.9 Resolution of the Research Questions . . . 184
6.10 Physically Constrained Verification . . . 185
6.11 Implications of the Research . . . 185
6.12 Limitations of the Study . . . 186
6.12.1 Validity and Reliability Considerations . . . 187
6.13 Discussion Summary . . . 189
6.14 Concluding Perspective . . . 189
6.15 Synthesis of Research and Fulfilment of Objectives . . . 190
6.16 Key Contributions to Knowledge and Practice . . . 192
6.17 Future Research Directions . . . 194
6.18 Concluding Remarks . . . 195

References **197**

List of Figures

1.1 Estimated 2030 economic value of Internet of Things adoption (Chui, Collins, and Patel, 2021) 2

1.2 Smart Home Market Size and Share: Industry Report, 2030 (Grand View Research, 2024b) 3

1.3 Smart Home Market Size and Share: Industry Report, 2030 (2) (Grand View Research, 2024b) 4

2.1 PRISMA-Style Flow Diagram of Study Selection 27

3.1 High-Level Architecture of the Continuous Security Assurance (CSA) Framework 74

3.2 Profiling-Grounded Anomaly Detection Workflow 77

3.3 IDS Classification Flow Diagram 78

3.4 CSA Monitoring Loop 80

3.5 The testing network containing the Raspberry Pi devices and associated peripherals 81

5.1 Overall, Random Forest was the best performer with an overall F1-score of 0.96 and accuracy of 96.42%. The average F1-scores were Normal (1.00), DoS (0.96), Spoofing (0.89), Tampering (0.89), and Information Disclosure (0.97) for each class. 137

5.2 The one-vs-rest implementation of the Random Forest (RF) classifier for SmartHome-StrideDS has been plotted based on an 80/20 training-testing split. The AUC for the classifier was micro-average 0.9984 and 0.9933 - 1.0000 for each class respectively. 138

5.3 One-vs-rest precision-recall curves for the Random Forest classifier when 80/20 train-test split is used for SmartHome-StrideDS. Overall, the classifier had an average-precision of 1.0000 for Normal, 0.9848 for DoS, 0.9511 for Spoofing, 0.9530 for Tampering and 0.9923 for Information Disclosure. 139
5.4 CSA dashboard during normal operation. 146
5.5 CSA dashboard during a coordinated STRIDE attack. 147
5.6 CSA dashboard during an adaptive adversarial attack. 148
5.7 RF F1-scores for the synchronised Hybrid Sys-Net Net-Host and Host-Only configurations on SmartHome-StrideDS per class. 148
5.8 Recall (Unsupervised) Per-Attack of hybrid SmartHome-StrideDS configuration and TON-IoT Linux process configuration. 150
5.9 Overall percentage of attacks that succeed against SmartHome-StrideDS for different perturbation budgets for RF_vanilla and RF_advtrain before and after CSA consistency verification. 161
5.10 Attack success rate across four perturbation budgets for RF_advtrain for both overall and post-verification for the TON-IoT host-process dataset. 162
5.11 Attack success rate for overall ML-ASR of RF_advtrain on TON-IoT network-flow dataset for the four perturbation budgets evaluated. 163
5.12 CSA physical-consistency gate activation frequencies across the evaluated SmartHome-StrideDS and TON-IoT adversarial configurations. 164
5.13 Adversarial success rates (ASRs): comparison of classifier-level adversarial success rate (ML-ASR) against the post-verification adversarial success rate (CSA-ASR) for the eight adversarial configurations tested at the highest budget tested. 166

List of Tables

1.1 Selected CVEs and their severity in residential IoT contexts 4

1.2 Additional CVEs referenced in the IoT threat landscape 5

2.1 Inclusion and Exclusion Criteria for the Systematic Literature Review . . 25

2.2 Comparative Analysis of Profiling Approaches in Security Assurance . . . 40

2.3 Analysis of Anomaly Detection Approaches in Smart-Home Security . . . 50

2.4 Comparative Analysis of Black-Box and Explainability-Oriented Machine Learning Techniques in IoT Anomaly Detection 52

2.5 Comparative Analysis of IDS and STRIDE-Oriented Approaches 63

3.1 Comparison of Threat-Modelling Approaches 85

4.1 Attack Scenarios and MITRE ATT&CK Mappings 97

4.2 Feature Rationale and STRIDE Attack Indications 101

5.1 Feature Statistics (Mean ± Std. Dev.) by Class 134

5.2 Traditional Global Threshold vs. Proposed Per-Device Hybrid Profiling . 134

5.3 Per-class F1-score and accuracy comparison of candidate classifiers on SmartHome-StrideDS (Synchronised Hybrid Sys-Net configuration). . . . 137

5.4 Comparison of relevant supervised, unsupervised/anomaly-detection and hybrid host–network approaches with the proposed framework; differing datasets and evaluation procedures mean that this is not a direct leaderboard. 141

5.5 Quantitative Comparison of Modalities Against STRIDE Attacks (Random Forest F1-Scores) 147

5.6 Binary Anomaly Detection Performance (Hybrid Sys-Net Dataset) 149

5.7 Per-Attack Recall (%) — Hybrid Sys-Net vs. TON Linux Dataset 150

5.8 Cross-dataset binary anomaly-detection performance comparison between TON-IoT network-only and Hybrid Sys-Net CSA configurations. 151

5.9 Forensic breakdown of adversarial evasion and CSA physical detection across all twenty-five evaluated examples (five per class). Gates: `stride` = S_{STRIDE}; `corr` = C_{corr}; `src_if` = source-class IF; `norm_if` = normal-class IF. 153

5.10 Oblivious attacker: ML-ASR and CSA-ASR (%) for RF_vanilla and RF_advtrain across all perturbation budgets, SmartHome-StrideDS ($n = 1{,}000$ per STRIDE class, $N_{\text{iter}} = 1{,}000$). Δ = ML-ASR − CSA-ASR (percentage points). 160

5.11 TON-IoT host-process dataset: ML-ASR / CSA-ASR (%) per attack class across all perturbation budgets, RF_advtrain, $N_{\text{iter}} = 300$. 162

5.12 TON-IoT network-flow dataset: ML-ASR / CSA-ASR (%) per attack class, RF_advtrain, $N_{\text{iter}} = 300$. 163

5.13 CSA gate activation frequency across all experimental configurations. n_{caught} = total intercepted samples per configuration. Percentages computed against n_{caught}. **Bold**: gate fires in $> 80\%$ of caught cases. . . 164

5.14 Consolidated adversarial robustness results across all experimental configurations at maximum evaluated ε. ML-ASR = classifier evasion rate; CSA-ASR = residual evasion after two-layer defence; Blocked = absolute count of intercepted post-ML-success attacks; Δ = CSA absolute reduction (pp). 165

5.15 Contextual comparison with published adversarial-robustness approaches. External results are retained in the metrics reported by their original publications; NR = not reported in a directly comparable form. 168

5.16 Numerical Comparison of Adversarial Attack Success Rates Before and After CSA Verification 170

6.1 Resolution of Research Questions 184

Chapter 1

Introduction

1.1 Background and Motivation

Over a relatively short period, the Internet of Things (IoT) has moved from a niche research concept to the operational backbone of a remarkable range of economic and social systems. Industrial automation, remote health monitoring, smart energy grids, and vehicle-to-infrastructure communication all now depend on dense networks of embedded sensing and actuation devices whose collective output drives decisions once reserved for human operators (Stoyanova et al., 2020; Mariya Ouaissa and Mariyam Ouaissa, 2025). The economic weight of this transformation is difficult to overstate: projections place the annual contribution of IoT-enabled systems at USD 5.5 to 12.6 trillion by 2030, with approximately 39 billion connected endpoints active at that date (Grand View Research, 2024b; Chui, Collins, and Patel, 2021).

The security investment that accompanies this expansion tells its own story. Global IoT security spending exceeded USD 35 billion in 2024 and is projected to surpass USD 140 billion by 2030, driven by escalating adversarial activity, tightening regulation, and organisational pressure to sustain operational resilience (Grand View Research, 2024a). Regulatory frameworks including NIST Cybersecurity Framework 2.0 (Pascoe, Quinn, and Scarfone, 2024) and the IoT Security Foundation Assurance Framework (IoT Security Foundation, 2021) now explicitly require continuous, evidence-based security monitoring rather than periodic compliance checks. Yet despite these pressures, significant implementation gaps remain, particularly in consumer-grade residential

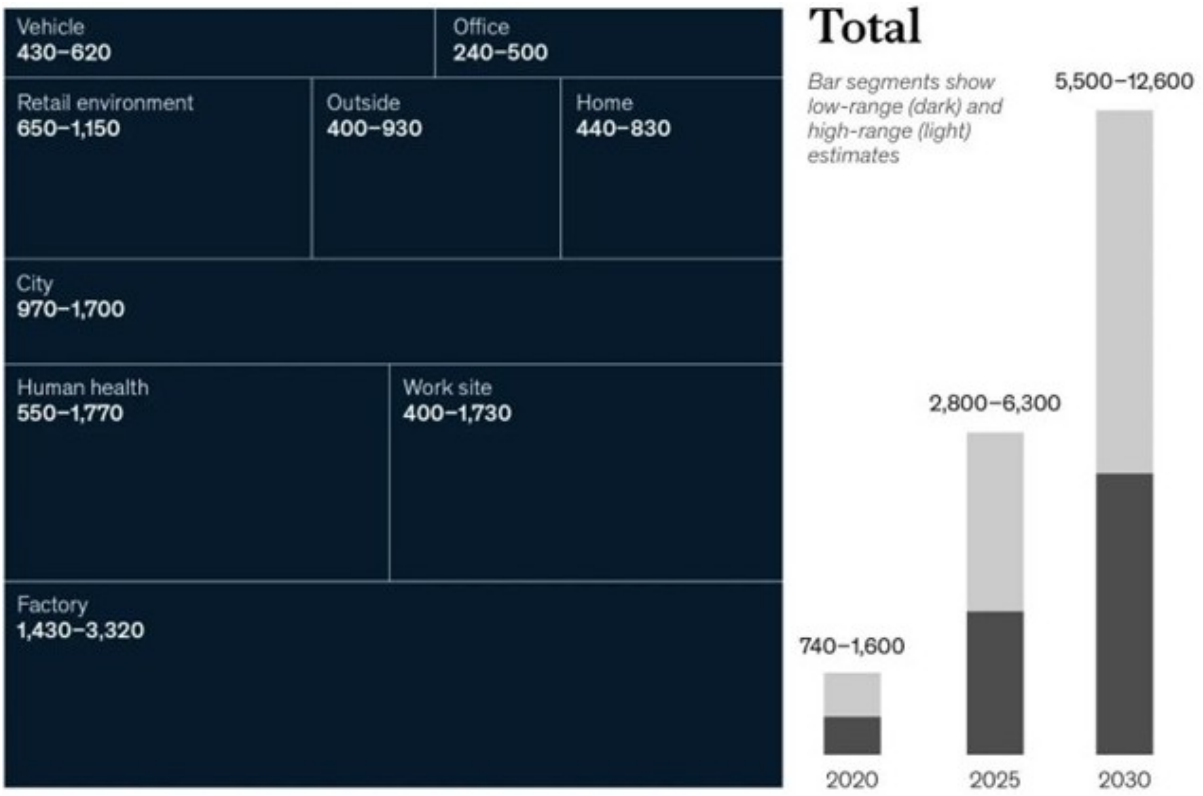

Figure 1.1: Estimated 2030 economic value of Internet of Things adoption (Chui, Collins, and Patel, 2021)

environments where neither professional security management nor enterprise-grade governance is present (Touqeer et al., 2021; Kabir, Gope, and Mohanty, 2023).

Those gaps have real operational consequences. Incident reports consistently document automated exploitation campaigns, large-scale botnet recruitment, and service-availability attacks targeting IoT endpoints (Yoachimik and Pacheco, 2025; Gould, 2025; Federal Bureau of Investigation, 2025). The deeper problem is not simply that vulnerabilities exist but that they are nearly impossible to track under partially observable, continuously evolving operational conditions (Watanabe et al., 2024; A. Shukla et al., 2022). An environment where devices are added, reconfigured, and updated without central visibility cannot be secured by mechanisms that presuppose a stable, well-defined perimeter.

The smart home is the sharpest expression of this challenge (Vardakis et al., 2024). A typical residence now contains fifteen to twenty networked devices, cameras, ther-

mostats, appliances, voice assistants, media hubs, none of which carries a professional security mandate and most of which run unpatched firmware on hardware not designed with security as a primary constraint (Canavese et al., 2024; Vardakis et al., 2024; Al-Salem, Almaiah, and Lutfi, 2023). The market for such deployments is substantial: the smart home sector was valued at USD 127.8 billion in 2024 and is forecast to reach USD 537.3 billion by 2030 (Grand View Research, 2024b).

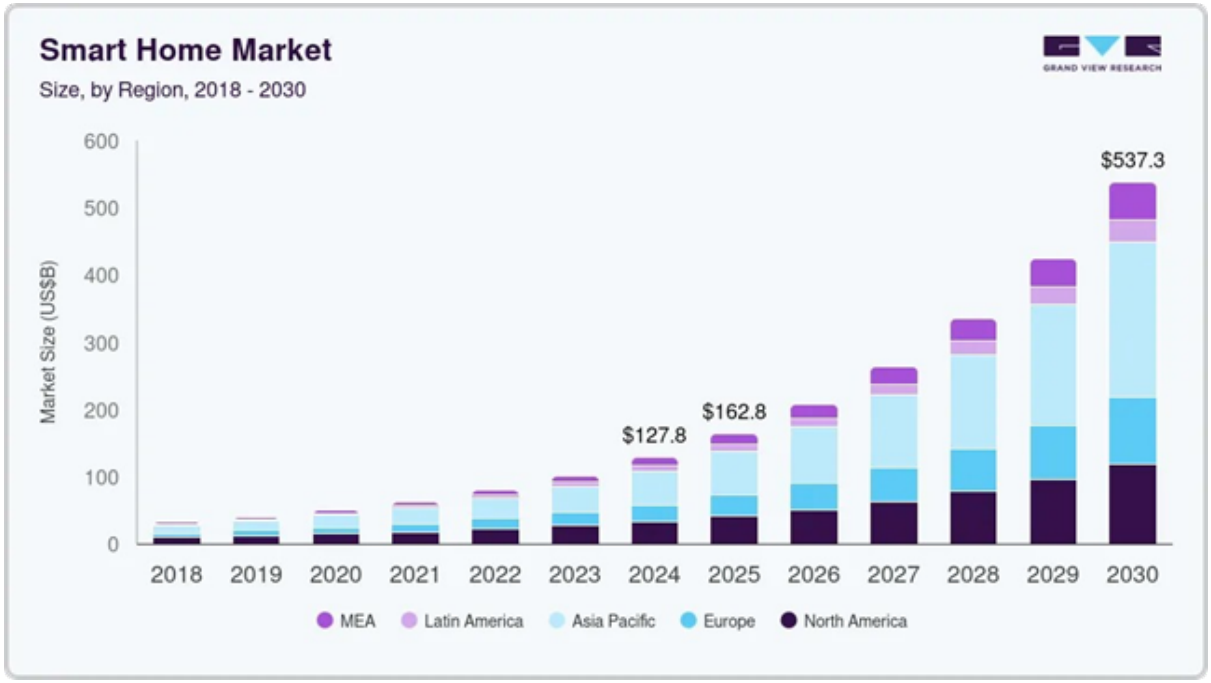

Figure 1.2: Smart Home Market Size and Share: Industry Report, 2030 (Grand View Research, 2024b)

The Bitdefender and NETGEAR 2025 IoT Security Landscape Report (Bitdefender and NETGEAR, 2025) quantifies the threat environment facing this population: over 6.7 million monitored smart homes and 58 million devices experienced, on average, 29 attacks per day. Recent incidents span the full range of attack categories. A 13,000-device botnet generated a 5.6 Tbps distributed denial-of-service attack (Yoachimik and Pacheco, 2025). A critical zero-day vulnerability in Edimax IC-7100 IP cameras (CVE-2025-1316) was disclosed (National Institute of Standards and Technology, 2025). The PUMABOT malware family targeting embedded Linux IoT devices emerged (Gould, 2025). The FBI issued warnings regarding BADBOX malware pre-installed in over twenty Android device models (Federal Bureau of Investigation, 2025). The US Cyber Trust Mark labelling

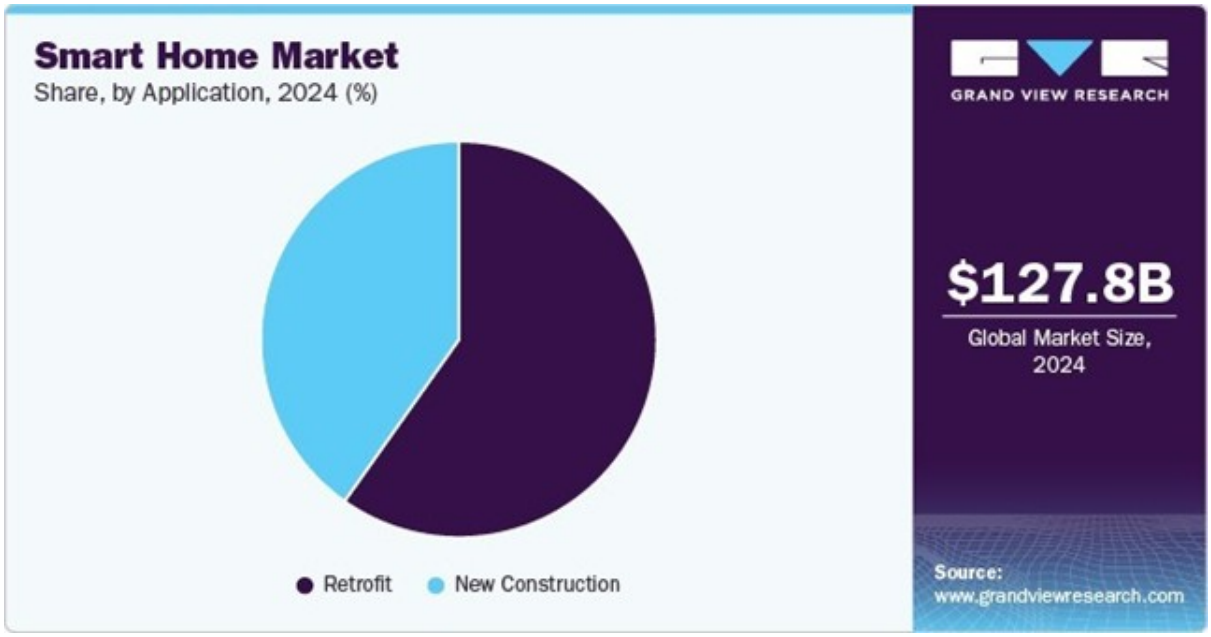

Figure 1.3: Smart Home Market Size and Share: Industry Report, 2030 (2) (Grand View Research, 2024b)

initiative was established (Federal Communications Commission, 2024). Table 1.1 and Table 1.2 document selected CVEs of relevance to the residential IoT environment examined in this thesis.

Table 1.1: Selected CVEs and their severity in residential IoT contexts

CVE	**Description (summarised)**	**CVSS**	**Affected component**
CVE-2025-37803	Linux kernel buffer-size overflow in `udmabuf_create()`	7.8 (High)	Multiple Linux kernel versions prior to patched releases
CVE-2025-37838	Linux kernel use-after-free in HSI `ssi_protocol` driver	7.8 (High)	Kernel versions 6.7 up to <6.12.24
CVE-2025-21751	Linux kernel `net/mlx5` error-flow mishandling	7.8 (High)	mlx5 kernel network driver

Table 1.2: Additional CVEs referenced in the IoT threat landscape

CVE	**Context**
CVE-2025-1316	Critical zero-day in Edimax IC-7100 IP cameras
CVE-2024-27130	QNAP Photo Station RCE in NAS ransomware campaigns
CVE-2017-5174	Bot.Genua RCE attack vector
CVE-2019-2725	Bot.Genua RCE attack vector
CVE-2018-17173	Bot.Genua RCE attack vector
CVE-2014-8361	Bot.Genua RCE attack vector

These incidents confirm that smart-home IoT has shifted from an emerging risk category to an active, high-volume operational threat surface (Spadafora, 2019; Tait, 2023). Per-device security agents are impractical given the resource constraints of most residential IoT hardware, and router- or network-layer protection addresses only a subset of the attack vectors actually observed in practice (Canavese et al., 2024; Job and Paul, 2021). Encrypted communications, adversarial behaviour embedded within legitimate processes, and living-off-the-land (LotL) techniques collectively undermine signature-based detection (A. Shukla et al., 2022; Lo et al., 2026). The structural flaws in existing assurance paradigms run deeper than any particular implementation shortcoming (Angermeir et al., 2024; Asonze et al., 2024).

What the field requires is an approach in which security is treated as a dynamic, continuously quantified property rather than a binary state periodically reasserted (Ardagna et al., 2017; Imran et al., 2026). Continuous Security Assurance (CSA) operationalises that requirement: it gathers and combines heterogeneous telemetry, measures deviations from established behavioural norms, and maintains an evidence-based security posture in near-real-time. This thesis develops a CSA framework specifically designed for the residential IoT environment, addressing the structural gap that motivates it.

1.1.1 Motivation for a Unified CSA Framework

Individual components of smart-home security have attracted sustained research attention. Authentication management (Vardakis et al., 2024), device-level monitoring (Chen et al., 2023), and network-traffic analysis (Farhan et al., 2025; Ben Slimane, Abd-Elkawy, and Maqbool, 2024) have each been studied in depth, but these contributions have been developed in isolation from one another. No existing methodology provides an evidence-based framework that combines system-level and network-level monitoring in a single, continuously operating assurance pipeline (A. Shukla et al., 2022).

The dynamic character of residential IoT environments, device churn, firmware upgrades, evolving adversarial tactics, compounds the difficulty of relying on siloed mechanisms (Vardakis et al., 2024; Sun et al., 2024). A unified CSA model must be capable of analysing device and network behaviour in real time (Lo et al., 2026; Softcat, 2024), consolidating evidence across both domains to form a coherent picture of security posture (Lo et al., 2026), and alerting operators to threats that span both planes, including covert intrusions that generate no single decisive indicator in either one alone (Lo et al., 2026; Chen et al., 2023). This thesis is a direct response to that unmet need.

1.2 The Need for Continuous Visibility in Smart-Home Security

Point-in-time security assessment is a poor fit for an environment that changes continuously. Smart-home devices are added, reconfigured, and removed without central oversight; firmware is updated asynchronously; cloud-service dependencies introduce behavioural changes that no on-premises monitor can anticipate (Watanabe et al., 2024; Meidan et al., 2023). Each of these dynamics expands the attack surface and increases the probability that a latent vulnerability will be exploited before it is noticed (Canavese et al., 2024; Watanabe et al., 2024).

A standard smart home contains fifteen to twenty networked devices, a number that continues to grow with the adoption of connected comfort systems, personal assistants, and domestic automation platforms (Buil-Gil et al., 2023). Each additional device brings with it heterogeneous communication protocols, varying authentication schemes, and distinct operational rhythms, all of which complicate the identification of security-relevant departures from normal behaviour (Meidan et al., 2023; Sivanathan et al., 2019; Yuan et al., 2023).

Security assurance that was accurate at one moment degrades rapidly in these environments (A. Shukla et al., 2022). Zero-day threats, continuously updated adversarial tooling, and the growing adoption of encrypted communications collectively reduce the visibility of malicious activity to classical intrusion detection systems (Khraisat and Alazab, 2021; Alhajjar, Maxwell, and Bastian, 2021; Papadogiannaki and Ioannidis, 2021). Signature-based, payload-based, and static-threshold approaches all struggle when the ground truth of what constitutes normal behaviour is itself shifting (Watanabe et al., 2024; Khraisat and Alazab, 2021; Taherdoost, 2023).

What sustained monitoring in this environment must account for is a set of operational attributes that conventional IT security tooling was not designed to handle. IoT endpoint churn, the regular addition, removal, and replacement of devices (Vardakis et al., 2024), means that no fixed device inventory can be assumed. Asynchronous firmware patches may themselves introduce new vulnerabilities (Bakhshi, Ghita, and Kuzminykh, 2024). Configuration drift, in which device and network settings shift through uncoordinated administrative actions (Robertson and Kosinski, 2026), creates a moving baseline that threshold-based detectors cannot track. Behavioural anomalies that manifest at the device layer, in CPU loading, memory consumption, or process activity, may never generate a network packet (Pang et al., 2021; Chen et al., 2023), leaving network-facing detectors entirely blind to them. Conversely, subtle changes in traffic patterns or protocol usage can signal malicious intent without producing any observable deviation in host-level metrics (Farhan et al., 2025).

Effective security assurance across this landscape requires persistent, multi-modal observation. Snapshots, periodic audits, and reactive alert systems do not capture temporal attack chains or the stealthy LotL strategies that characterise modern adversary behaviour (A. Shukla et al., 2022; Lo et al., 2026). The inability to measure, correlate, and interpret host- and network-level data in near-real-time creates windows during which attackers can escalate privileges, exfiltrate data, or degrade service availability without triggering any alert (Lo et al., 2026). Addressing this visibility deficit is the operational imperative that drives the present work (Lo et al., 2026).

1.3 The Sys-Net Blind Spot in Smart-Home Continuous Security Assurance

The dominant approach to smart-home security monitoring is network-centric. Traffic flows are analysed and anomalies are flagged against signature libraries or statistical baselines derived from packet-level observations (Khraisat and Alazab, 2021; Farhan et al., 2025). This approach is effective for threats whose primary observable artefacts reside at the network layer but systematically fails for attacks that produce their most significant effects at the host level. A malicious process consuming CPU and memory, an unauthorised modification to a configuration file, or a privilege escalation exploiting a firmware vulnerability may generate no discernible network signature whatsoever (Chen et al., 2023). From the perspective of a network-facing IDS, the device appears entirely normal.

The mirror failure occurs in host-only monitoring. ARP-based man-in-the-middle attacks, botnet recruitment via protocol-compliant traffic, and covert command-and-control communications all manipulate the network layer without disturbing host-level metrics. A host-only monitor sees nothing unusual; the device's CPU, memory, and process state remain within normal bounds.

This study designates the combined consequence of these two blind spots the **Sys-Net Blind Spot**: the structural incapacity of single-modality monitoring, whether host-only or network-only, to observe security-relevant behaviour that is jointly encoded across both telemetry planes (Lo et al., 2026; Chen et al., 2023). The blind spot has four operationally significant consequences.

The first is incomplete threat detection. Network-centric IDS solutions cannot detect system-level attacks; host-centric solutions cannot detect network-layer manipulation. Attackers who understand this architecture can design campaigns that remain invisible to any single-plane detector (Pang et al., 2021; Lo et al., 2026).

The second consequence is ambiguous behavioural interpretation. An anomaly observed in isolation, whether a CPU spike, an unusual outbound connection, or an elevated memory reading, is insufficient to establish malicious intent. Benign causes such as firmware updates, background synchronisation tasks, or temporary resource contention produce identical single-plane signatures (Meidan et al., 2023). Only when host-level and network-level signals are correlated can many of these ambiguities be resolved.

Third, security dashboards that rely exclusively on network indicators present operators with a partial and potentially misleading picture of device activity. The operational risk is overconfidence: an apparently clean network feed does not imply a secure device (Chen et al., 2023; Lo et al., 2026).

Fourth, the absence of cross-layer visibility extends the interval between compromise and detection, giving attackers a window in which to escalate privileges and establish persistence before any alert fires (Lo et al., 2026). This thesis focuses on detection and interpretation; addressing the response latency problem that follows from detection is identified as future work.

Closing the Sys-Net Blind Spot requires an architecture capable of collecting and harmonising heterogeneous telemetry from both data planes in near-real-time (Lo et al., 2026; Chen et al., 2023), aggregating that evidence to disambiguate signals that would be

misclassified under isolated monitoring (Chen et al., 2023), mapping the resulting anomalies to structured threat semantics that support informed operator decisions (Mariya Ouaissa and Mariyam Ouaissa, 2025; Shevchenko et al., 2018), and maintaining an accurate security posture that updates continuously as new telemetry is processed (Imran et al., 2026). A framework satisfying all four requirements transforms smart-home security assurance from a network-limited monitoring activity into a fully integrative, host-aware, evidence-based discipline (Imran et al., 2026).

1.4 Problem Statement

Existing smart-home IoT security approaches share a structural property that limits their operational value: they are predominantly fragmented and network-centric, directing attention either toward device-level monitoring or toward network traffic inspection but not toward both in combination (Lo et al., 2026; Chen et al., 2023). They tend to rely on episodic or partial observations rather than continuously aggregating and validating cross-domain evidence (A. Shukla et al., 2022).

In this thesis, an evidence-based approach to security assurance refers specifically to a methodology that integrates, correlates, and analyses both host-level and network-level telemetry to produce a unified, interpretable representation of the smart-home security state (A. Shukla et al., 2022; Lo et al., 2026). The absence of such an approach leaves structural blind spots that sophisticated adversaries can deliberately exploit (A. Shukla et al., 2022; Lo et al., 2026).

The operational consequences of this gap are concrete. Without sustained collection and correlation of cross-layer telemetry, security operators cannot track subtle host-level anomalies alongside network-layer deviations, which substantially reduces situational awareness (Lo et al., 2026). Coordinated or stealthy adversarial behaviour, including LotL intrusions that produce minimal observable network traffic, cannot be

identified with adequate reliability (Lo et al., 2026). Security decisions are necessarily based on point-in-time snapshots rather than continuously quantified evidence, making sustained enforcement of security posture over time impracticable (A. Shukla et al., 2022; Angermeir et al., 2024). And configuration drift, emerging threat patterns, and device-level vulnerability exposure are identified later than they need to be, extending the window between compromise and detection (Pascoe, Quinn, and Scarfone, 2024).

The lack of effective hybrid evidence integration is the technical core of the problem. When CPU load spikes, memory anomalies, and abnormal process execution co-occur with covert network-layer activity, a system that monitors each plane independently cannot establish the causal relationship between them, cannot reliably infer threat intent, and cannot maintain a continuously updated security state (Lo et al., 2026; A. Shukla et al., 2022).

Episodic assessment compounds the difficulty. Security posture shifts continuously as a result of device churn, configuration drift, and the adaptation of adversarial techniques (Watanabe et al., 2024). Without longitudinal tracking, exposure windows arise in which emerging weaknesses remain undetected. A continuous measurement framework would maintain the operational security state as a live, interpretable, and actionable quantity (Imran et al., 2026).

This study reframes the problem beyond the classical anomaly detection question of whether an event is suspicious. The question posed here is broader: whether the system as a whole is operating within its expected security envelope, and whether that envelope is being maintained in a way that is continuously evidenced, semantically interpretable, and resilient to deliberate evasion. The three operational commitments that follow from this reframing are systematic accumulation of heterogeneous Sys-Net evidence (Lo et al., 2026), live analysis of behavioural anomalies mapped to the STRIDE threat taxonomy (Shevchenko et al., 2018; Mariya Ouaissa and Mariyam Ouaissa, 2025), and a continuously renewed security posture measure (Imran et al., 2026). Meeting all three simultaneously is the research problem this thesis addresses.

1.5 Research Aim, Objectives, and Research Questions

Research Aim

The principal aim of this research is to design and develop a hybrid Continuous Security Assurance framework that fuses device-level and network-level security measurements through behavioural profiling, anomaly detection, and STRIDE-aligned intrusion detection mapping, enabling real-time threat identification and continuous security posture assessment in IoT-enabled smart-home environments.

Research Questions

Research Question 1

How can hybrid behavioural profiling be architecturally and statistically integrated with anomaly detection and behaviourally validated IDS classification to realise Continuous Security Assurance in heterogeneous smart-home environments?

Research Question 2 — Hybrid Profiling Foundation

How can hybrid device-level and network-level telemetry be statistically modelled to establish stable behavioural baselines for continuous security quantification in smart homes?

Research Question 3 — Profiling-Grounded Anomaly Detection

To what extent does profiling-grounded anomaly detection improve anomaly identification compared to single-modality, device-only or network-only, approaches?

Research Question 4 — Continuous Security Assurance Evaluation

What are the statistical, architectural, and operational implications of deploying a real-time hybrid profiling framework for Continuous Security Assurance in smart-home systems?

Research Objectives

Four objectives structure the programme of work.

Objective 1 is to design and develop a hybrid system–network behavioural profiling framework, supported by the construction of the SmartHome-StrideDS dataset, that systematically captures synchronised host-level and network-level telemetry and enables the statistical modelling of stable behavioural baselines for Continuous Security Assurance in IoT-enabled smart homes (Gavric, Prasad Bhandari, and Shalaginov, 2026; Sivanathan et al., 2019).

Objective 2 is to design and develop a hybrid machine learning-based behavioural anomaly detection model that leverages the statistically grounded hybrid Sys-Net profiling foundation to detect and quantify deviations from established normal behaviour, thereby improving detection accuracy in heterogeneous smart-home environments (Lara et al., 2023; Pang et al., 2021).

Objective 3 is to design and implement a multi-class intrusion detection mechanism aligned with STRIDE-based threat categorisation, enabling intent-aware classification and contextual threat analysis across heterogeneous IoT devices within the hybrid system–network framework (Mariya Ouaissa and Mariyam Ouaissa, 2025; Shevchenko et al., 2018).

Objective 4 is to design, validate, and empirically evaluate the adversarial robustness of a unified Continuous Security Assurance framework that integrates hybrid

profiling, profiling-grounded anomaly detection, and STRIDE-aligned intrusion detection into a cohesive real-time security posture quantification model, including assessment of its resilience against bounded feature-space perturbations under both oblivious and adaptive threat models (Goodfellow, Shlens, and Szegedy, 2015; Vassilev et al., 2025; Biggio and Roli, 2018).

1.6 Scope

This thesis addresses threat detection in IoT-enabled smart-home environments, with emphasis on achieving continuous, interpretable, and adversarially robust security assurance through hybrid system-network behavioural analysis. The work is deliberately bounded at the detection and assurance layer. Automated mitigation, incident response, and remediation fall outside its remit, on the principled grounds that any meaningful response presupposes faithful detection, and that achieving such detection in a heterogeneous Sys-Net environment is itself a substantive and independently unresolved scientific challenge (A. Shukla et al., 2022; Lo et al., 2026).

Threat coverage is scoped to four of the six STRIDE categories: Denial of Service, Spoofing, Tampering, and Information Disclosure. These represent the attack classes most directly observable through synchronised host-level and network-level telemetry in residential IoT deployments (Shevchenko et al., 2018; Mariya Ouaissa and Mariyam Ouaissa, 2025). The remaining two categories, Repudiation and Elevation of Privilege, are intentionally excluded from empirical evaluation. Repudiation threats primarily manifest through audit log manipulation, which lies outside the telemetry streams captured by the SmartHome-StrideDS dataset. Elevation of Privilege, while theoretically mappable to system-level indicators, requires access to privilege and permission metadata not consistently available across the heterogeneous device population of the testbed. Both exclusions are acknowledged as limitations and identified as directions for future work.

The adversarial robustness of the detection layer is additionally examined within this scope, specifically to verify that the CSA consistency verification layer maintains operational integrity against an informed adversary applying bounded feature-space perturbations, consistent with the assurance goals of the framework and the realistic threat conditions of smart-home deployments (Goodfellow, Shlens, and Szegedy, 2015; Biggio and Roli, 2018; Vassilev et al., 2025).

1.7 Thesis Contributions

The thesis makes five interconnected contributions to the IoT security literature.

Hybrid Sys-Net Profiling

Modern IoT intrusions increasingly adopt stealth-oriented techniques, LotL host manipulation, protocol-compliant network abuse, pre-exfiltration preparation, that are designed to minimise distinctive artefacts in either telemetry plane alone (Lo et al., 2026). Single-modality profiling approaches exhibit structural blind spots precisely in these scenarios: malicious activity that preserves plausibility within one observation domain evades any detector that examines only that domain (Lo et al., 2026). This problem is particularly acute in smart homes, where heterogeneous device roles and bursty user-driven traffic create telemetry patterns that closely resemble adversarial behaviour (Meidan et al., 2023).

The profiling layer proposed in this work addresses the problem not by introducing new metrics but by changing how normal operation is defined. Benign device behaviour is modelled by jointly capturing host-level telemetry and network-level behaviour over time, so that normality is expressed as cross-layer operational consistency rather than as a threshold applied to any single indicator. This formulation reflects recent calls for context-aware and relational modelling in IoT security research (Meidan et al., 2023; Lara et al., 2023; Lo et al., 2026).

The contribution is grounded in the empirical observation that system-level and network-level behaviours in smart-home environments are not independent. Smart-home hubs and gateways operate as cyber-physical control points where network interactions directly influence resource utilisation and internal state (Gavric, Prasad Bhandari, and Shalaginov, 2026), so that

$$P(H, N) \neq P(H)\, P(N) \tag{1.1}$$

where H denotes system telemetry and N denotes network telemetry. Modelling the joint distribution rather than the marginals significantly improves the separability of benign from adversarial behaviour (Chen et al., 2023). Across all four implemented attack scenarios, the hybrid profiling layer exposes attacks that intentionally suppress artefacts within isolated planes by enforcing cross-layer operational consistency (Lo et al., 2026; Chen et al., 2023).

Profiling-Grounded Anomaly Detection

The anomaly detection layer proposed here treats deviations not as discrete binary events but as continuous, measurable quantities grounded in the per-feature mean and standard deviation established by the profiling layer. This framing enables real-time quantification of behavioural drift with interpretable statistical semantics, supporting the interpretability and continuity requirements of the CSA framework (Ardagna et al., 2017; Samek, Montavon, Lapuschkin, et al., 2021). Where existing approaches either lack behavioural grounding (Sommer and Paxson, 2010), operate on single-plane telemetry (Pang et al., 2021), or provide limited explainability (Samek, Montavon, Lapuschkin, et al., 2021), the proposed layer provides all three properties through its direct connection to the profiling baseline.

STRIDE-Aligned Intrusion Detection

The intrusion detection layer operates downstream of profiling-grounded anomaly detection, classifying only events that have already cleared behavioural normality thresholds. This architectural separation keeps the classifier's input distribution restricted to genuinely security-relevant state transitions, improving precision and reducing the false-positive burden that plagues IDS systems trained on unfiltered telemetry (Meidan et al., 2023; Sommer and Paxson, 2010). A STRIDE-aware Random Forest classifier maps validated anomalies to specific threat categories, producing alerts that carry structured threat semantics rather than undifferentiated alarm signals (Shevchenko et al., 2018; Mariya Ouaissa and Mariyam Ouaissa, 2025).

SmartHome-StrideDS Dataset

Beyond its methodological contributions, the thesis delivers SmartHome-StrideDS, a purpose-built dataset that provides fully synchronised host-level and network-level observations at each time step, enabling direct modelling of Sys-Net relationships without post-hoc fusion overhead. Several widely used IoT security datasets are network-centric rather than providing time-synchronised host–network observations; BoT-IoT, for example, is built from labelled network traffic (Koroniotis et al., 2019). The attack classes in SmartHome-StrideDS are operationalised through STRIDE-aligned scenarios using legitimate system utilities and protocol-compliant traffic, so that no attack can be separated using single-plane artefacts alone. This design specifically targets the evaluation of hybrid profiling under realistic adversarial conditions.

Adversarial Robustness of the CSA Framework

A physically constrained adversarial robustness evaluation verifies that the CSA consistency verification layer maintains detection integrity under deliberate feature-space

evasion. Bounded perturbations are applied within a per-feature epsilon budget defined as a scaling factor on each feature's observed operational range (Papernot et al., 2017; Alhajjar, Maxwell, and Bastian, 2021). Both a standard Random Forest classifier and an adversarially hardened variant retrained on augmented data are evaluated against an oblivious attacker and an adaptive attacker whose loss function explicitly penalises violations of all three CSA consistency rules: temporal consistency (T_{EWMA}), causal correlation consistency (C_{corr}), and STRIDE semantic consistency (S_{STRIDE}) (Biggio and Roli, 2018; Vassilev et al., 2025). The evaluation is conducted across three independent datasets to confirm that the defence layer's properties transfer across heterogeneous IoT feature spaces (Moustafa, 2021). Empirical results confirm that the per-class Isolation Forest confirmation gate, combined with the three physical consistency rules, substantially reduces net attack success rates while maintaining a false positive rate below 0.2% on clean normal traffic.

1.8 Thesis Outline

Chapter 2 presents a critical review of the literature on smart-home IoT security, intrusion detection systems, and security assurance paradigms, identifying the Sys-Net blind spot as the primary research gap.

Chapter 3 describes the theoretical architecture of the proposed CSA framework, the methodological justification, the smart-home testbed, and the evaluation design.

Chapter 4 consolidates the design and implementation of hybrid behavioural profiling, SmartHome-StrideDS, profiling-grounded anomaly detection, the STRIDE-aligned IDS, consistency verification, and security-posture quantification.

Chapter 5 presents the experimental results and validation of the complete CSA framework, including baseline comparisons and adversarial robustness evaluation.

Chapter 6 provides a structured critical interpretation of the empirical findings, relates them to the research objectives and questions, summarises the contributions and limitations, and identifies directions for future work.

1.9 Chapter Summary

This chapter has established the foundational problem motivating the thesis. Smart-home IoT environments are dynamic, heterogeneous, and poorly served by security mechanisms designed for controlled enterprise settings. Static assurance models cannot keep pace with device churn, firmware updates, configuration drift, and the continuous evolution of adversarial techniques (A. Shukla et al., 2022; Watanabe et al., 2024). The dominant network-centric approach to continuous monitoring creates the Sys-Net Blind Spot, in which host-level compromises that generate no network signature pass entirely undetected (Chen et al., 2023; Lo et al., 2026). The research aim, four objectives, and four research questions that guide the remainder of the thesis were established as a direct response to this structural gap, and the scope of the empirical contribution was bounded at the detection and assurance layer.

Chapter 2

Literature Review

2.1 Introduction

The rapid proliferation of Internet of Things (IoT) technologies has fundamentally transformed homes into highly integrated cyber-physical ecosystems (Sovacool and Furszyfer Del Rio, 2020; Vardakis et al., 2024). Heterogeneous devices such as sensors, actuators, appliances, gateways, and cloud-based services are combined in smart homes to automate, remotely control, and make intelligent decisions (Sovacool and Furszyfer Del Rio, 2020; Bitdefender and NETGEAR, 2025). Although such connectivity provides substantial functional and social advantages (Grand View Research, 2024b; Sovacool and Furszyfer Del Rio, 2020), it also broadens the attack surface and creates difficult security threats (Vardakis et al., 2024; Yuan et al., 2023). The resulting environments are characterised by device diversity, limited computational resources, continuous execution, and dynamic behavioural patterns, making the design and testing of practical security mechanisms complex (A. Shukla et al., 2022; Touqeer et al., 2021).

In these settings, conventional cybersecurity methods that focus on perimeter protection or isolated incident-based intrusion detection are increasingly inadequate (Khraisat and Alazab, 2021; AlSalem, Almaiah, and Lutfi, 2023). Smart-home systems evolve continuously as a result of firmware updates, device additions and removals, changes in user behaviour, and network conditions (Watanabe et al., 2024; Meidan et al., 2023). Security must therefore be considered not only in terms of detecting discrete attacks, but through the lens of regularly assessing whether the system maintains an

acceptable security posture during operation (A. Shukla et al., 2022; Pascoe, Quinn, and Scarfone, 2024). This broader goal is known as **security assurance**, the systematic gathering of evidence to confirm that security requirements are met and functioning effectively across the lifecycle of the system (A. Shukla et al., 2022).

It is important to distinguish **security assurance** from **intrusion detection**, as the two concepts serve different purposes (A. Shukla et al., 2022; Khraisat and Alazab, 2021). Intrusion detection mechanisms primarily focus on identifying malicious activities or policy violations after or during their occurrence (Khraisat and Alazab, 2021; A. Shukla et al., 2022). Security assurance, by contrast, adopts a holistic perspective that evaluates the ongoing trustworthiness, resilience, and risk exposure of the system as a whole (A. Shukla et al., 2022; Pendleton et al., 2016). Detection is therefore one component of assurance rather than a complete solution. An assurance-driven approach integrates multiple complementary techniques to continuously observe system behaviour, identify deviations, interpret threats, and reason about their security implications (Ardagna et al., 2017; Lo et al., 2026).

Across the literature, three foundational technical components consistently underpin security assurance in IoT and smart-home environments (A. Shukla et al., 2022; Khraisat and Alazab, 2021). The first is **behavioural profiling**, which establishes baseline models of normal system or network activity to characterise expected operational patterns (Sivanathan et al., 2019; Ben Slimane, Abd-Elkawy, and Maqbool, 2024). The second is **anomaly detection**, which identifies statistically or semantically significant deviations from these baselines that may indicate faults, misuse, or adversarial behaviour (Pang et al., 2021; Aparcana-Tasayco, Deng, and Park, 2025). The third is **intrusion detection systems (IDS)**, which employ signature-based, anomaly-based, or hybrid strategies to detect and classify malicious events, often guided by structured threat models such as STRIDE to interpret adversarial intent (Mariya Ouaissa and Mariyam Ouaissa, 2025; Shevchenko et al., 2018). Together, these components form the core analytical mechanisms through which security-relevant evidence is generated (A. Shukla

et al., 2022).

Nevertheless, a substantial portion of the literature treats profiling, anomaly detection, and IDS as singular or event-driven processes, often evaluated within a static or periodic paradigm (A. Shukla et al., 2022; Watanabe et al., 2024). These techniques offer only a snapshot view and frequently fail to detect slow behavioural changes, subtle attacks, or gradual deterioration in security posture (A. Shukla et al., 2022; Aparcana-Tasayco, Deng, and Park, 2025). Such limitations have inspired a growing focus on Continuous Security Assurance (CSA), which emphasises long-lasting monitoring, adaptive modelling, and runtime verification of security evidence (Ardagna et al., 2017; Angermeir et al., 2024).

This chapter critically reviews the state of the art in smart-home IoT security assurance. It first examines classic models of profiling, anomaly detection, and intrusion detection, along with their architectural assumptions and constraints (Khraisat and Alazab, 2021; A. Verma and Ranga, 2020). It then examines new continuous assurance paradigms and associated methods (Ardagna et al., 2017; Angermeir et al., 2024). Finally, it summarises the identified literature gaps and motivates the need for a unified, multi-layer continuous security assurance framework (A. Shukla et al., 2022; Ardagna et al., 2017).

The chapter was developed as a narrative literature review but a systematic review protocol was used in the correction phase to validate the literature and provide a transparent and reproducible process of literature used for this chapter. The existing evidence base was organised by this protocol around the principal themes of this thesis, which are: smart-home IoT security, behavioural profiling, anomaly detection, intrusion detection, STRIDE-based threat interpretation, Continuous Security Assurance and adversarial robustness.

The initial (primary) review period was set as 2015 – 2025. Previous seminal publications were included individually when they served as sound methodological foun-

dations and literature that appeared after the identified systematic-review time frame were included as additional "recent" work documents outside the time frame included in the identified systematic review.

2.1.1 Review Questions

ID	Review question
LRQ1	What security-assurance approaches have been proposed for smart-home and IoT environments, and what limitations affect their suitability for continuous security monitoring?
LRQ2	How have behavioural profiling and anomaly-detection techniques been applied to host-level, network-level and hybrid system–network telemetry in smart-home IoT environments?
LRQ3	What intrusion-detection and STRIDE-aligned approaches have been proposed for identifying, classifying and interpreting threats in smart-home and IoT systems?
LRQ4	What limitations remain in existing approaches with respect to continuous security assurance, hybrid system–network visibility, security-posture quantification and adversarial robustness?

2.1.2 Information Sources and Search Strategy

The primary databases were Scopus, IEEE Xplore and the ACM Digital Library. Google Scholar was used as a supplementary discovery and citation-chaining source. Two thematic search expressions were employed.

Search	Theme	Search string
S1	Profiling, anomaly detection, IDS, security assurance and STRIDE	("smart home" OR "smart-home" OR "residential IoT") AND ("intrusion detection" OR "anomaly detection" OR "behavioural profiling" OR "behavioral profiling" OR "security assurance") AND ("hybrid intrusion detection" OR "host-based intrusion detection" OR "network-based intrusion detection" OR "host-network" OR "cross-layer" OR "security posture" OR STRIDE OR "threat modelling" OR "threat modeling")
S2	Adversarial robustness	("smart home" OR "smart-home" OR "residential IoT") AND ("intrusion detection" OR "anomaly detection") AND ("adversarial attack" OR "adversarial robustness" OR "evasion attack" OR "adversarial training")

The raw query executions produced the following search-result counts:

Database	S1	S2
Scopus	34	12
IEEE Xplore	50	23
ACM Digital Library	815	260

We conducted two thematic searches, and the total number of query-level retrieval occurrences was 1,194 in Scopus, IEEE Xplore, and the ACM Digital Library. The results reflect the number of times each study was retrieved, not the number of times each study was published, because the same study may have been retrieved more than once by a thematic search, and may have been indexed in more than one database. The number of raw retrievals is therefore presented to illustrate the wide scope and reach of the search strategy and not as the number of individual studies screened for inclusion.

Because the reported database counts represent query-level retrieval occurrences, duplicate appearances were expected where the same publication was returned by more than one thematic search or was indexed in more than one database. A total of 604 bibliographic records were captured for the retrospective validation. Duplicate bibliographic instances were consolidated by comparing publication title, authorship, year and, where available, DOI. Seventy-seven (77) duplicate records were removed, leaving 527 unique records. Each publication was therefore considered only once during the validation. The raw total of 1,194 is retained to report the coverage of the database searches and should not be interpreted as the number of unique records subjected to screening.

2.1.3 Eligibility Criteria

Table 2.1 summarises the eligibility criteria applied to the systematic literature review. The criteria cover the publication period, language, peer-review status, research relevance, methodological evidence, and eligible publication types.

Table 2.1: Inclusion and Exclusion Criteria for the Systematic Literature Review

Inclusion	**Exclusion**
Published during 2015–2025	Outside the review period, except seminal supporting literature
English language	Non-English
Peer-reviewed journal or full conference paper	Non-peer-reviewed material
Smart-home, residential IoT or directly transferable IoT security	No meaningful relevance to IoT security
Profiling, anomaly detection, IDS, assurance, STRIDE or adversarial robustness	Outside LRQ1–LRQ4
Sufficient methodological/technical evidence	Insufficient methodological evidence
Empirical, systematic-review or technically substantiated evidence	Purely contextual material with no direct contribution to the review questions
Additional exclusion criteria: Proceedings title/whole volume; Editorial; Conference abstract only; Poster; Preface/front matter; Presentation without a full peer-reviewed paper.	

The eighty-nine (89) works cited in Chapter 2 have been identified and validated by the author's retrospective validation. There are 12 publications that exceeded the prescribed period of primary review (2015–2025): 4 of which were earlier foundational studies and 8 publications were from 2026. The 12 studies selected were used as supporting literature, but did not make up the primary systematic-review set. The remaining 77 publications were screened at title and abstract level for direct relevance to LRQ1–LRQ4 and the eligibility criteria defined above. 9 publications were identified as supporting, not primary review, studies, which were mainly used for methodological foundations for the IoT research field, general background for the IoT, background for the dataset, background for the networks used, or concepts of explainability, general background for adversarial-learning methods or consultation of policy/contextual information. This resulted in a final systematic synthesis set of 68 studies that directly addressed the technical themes of the review.

The identification and screening processes used to identify studies for the SLR and determine whether they are eligible for inclusion are shown in Figure 2.1. The database searches yielded 1,194 retrievals, of which 604 bibliographic records were obtained and validated against the list of previous findings. After the duplicates were eliminated, 527 unique records remained. Of the 89 works included in the reference list in Chapter 2, 12 were outside the primary review period and nine were considered additional supporting literature, leaving 68 studies to be included in the systematic synthesis. The figure thus indicates the scope of the search information separately from the synthesis evidence base. These figures represent two distinct stages of retrospective validation rather than a single continuous screening sequence. The 527 unique records describe the deduplicated database-search records, whereas the 89 works represent the literature already cited in Chapter 2 and subsequently assessed for inclusion in the primary systematic synthesis. Therefore, the difference between 527 and 89 should not be interpreted as an additional number of excluded studies.

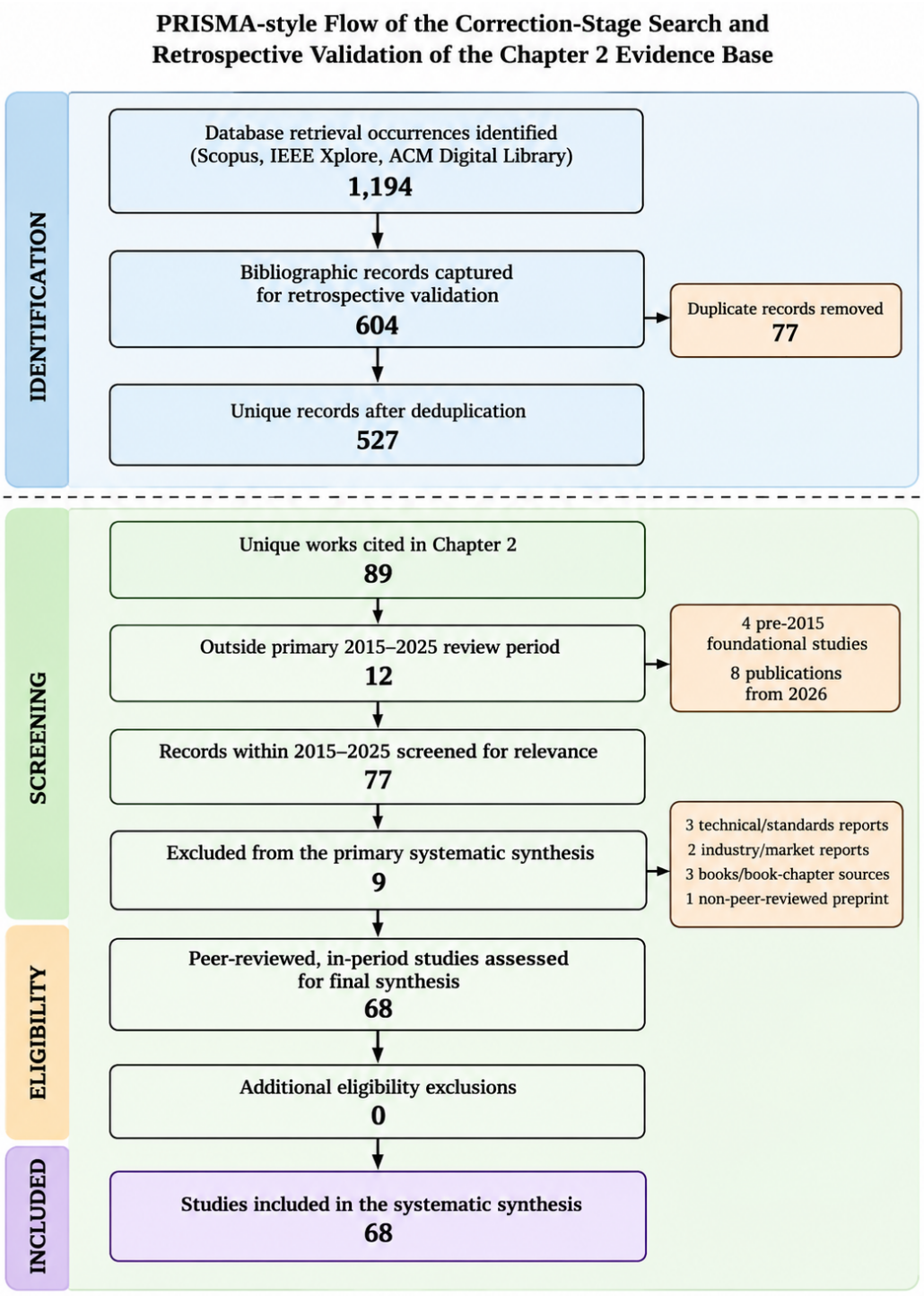

Figure 2.1: PRISMA-Style Flow Diagram of Study Selection

The literature was already substantively included in Chapter 2 and during the preceding relevance-screening stage sources were separated that were not directly relevant to the literature review questions, for this reason the retrospective validation was applied. Publications that were not used in the main synthesis were not included in the 68 primary

synthesis publications, except for methodological and contextual support.

2.1.4 Full-Text Screening, Quality Assessment and Data Extraction

Full-text screening. After screening for title and abstract, 68 publications were then selected for full-text screening for inclusion in the primary synthesis based on the defined eligibility criteria. The assessment was used to confirm relevance to substance within one or more of the Literature Review Questions [LRQ1, LRQ2, LRQ3 and LRQ4] and to ensure that the methods, evidence and findings reported were detailed enough for use in the review. There were no additional publications excluded at this stage, leaving 68 publications for the final systematic synthesis.

Quality assessment. The methodological quality was not quantified, but rather evaluated qualitatively. For each study, the extent to which the study met the following criteria were considered: peer review, clarity of the study method, sufficient integration of data/experimental evidence (for applicable studies), transparency of evaluation procedure and results, and a straightforward relevance to the technical themes of the review. Studies with inadequate methodological and technical evidence were not included in the primary synthesis. This assessment was not intended to compare retained publications to each other, but was a way to confirm their suitability.

Data extraction. The relevant information for each publication included in the primary synthesis was gathered and recorded in structured format and according to the review questions. The information retrieved included: publication data and information about the data or experimental setting, profiling or detection technique, machine-learning or analytical method if any, attack or threat coverage, relevance to LRQ1–LRQ4, limitations, evidence of performance. The extracted information was organised into themes and comparative analyses were made across specific themes throughout Chapter 2, based on these extracted characteristics.

2.2 Security Assurance in Smart-Home Environments

The security assurance of smart-home IoT ecosystems is made complex by the growing heterogeneity of devices, limited computing resources, and the requirement for continuous operation (Canavese et al., 2024; Kabir, Gope, and Mohanty, 2023). Devices range from simple sensors to complex actuators, each with different behavioural patterns and communication protocols. This heterogeneity makes it difficult to monitor and enforce security policies consistently (Vardakis et al., 2024; Canavese et al., 2024). Because these devices operate continuously and autonomously, and their behaviour varies over time, security assurance systems must be able to respond to temporal and contextual variations (Aparcana-Tasayco, Deng, and Park, 2025).

Security is further challenged by observability limitations. Many smart-home devices have limited telemetry reporting, making detailed profiling or anomaly identification difficult. Network traffic may be encrypted or intermittent, and device-level logs are sparse or inconsistent (Stoyanova et al., 2020; Papadogiannaki and Ioannidis, 2021). As a result, static and unchanging security assurance models tend to leave significant areas of the attack surface unobserved (A. Shukla et al., 2022).

2.2.1 Smart-Home Architecture and Threat Landscape

Smart-home systems are characterised by a wide range of communication protocols , including Wi-Fi, Zigbee, and Bluetooth Low Energy, between devices with varying processing capabilities and memory limitations (F. Hussain et al., 2020; Al-Fuqaha et al., 2015). This architectural diversity creates numerous potential attack vectors, including device compromise, network sniffing, and horizontal malware propagation (A. Shukla et al., 2022; Alrawi et al., 2019; Díez-Franco, García Bringas, and Ugarte-Pedrero, 2024).

The threat landscape of smart homes is also shaped by the absence of a universal security template. The STRIDE framework categorises threats as Spoofing, Tamper-

ing, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (Shevchenko et al., 2018; Mariya Ouaissa and Mariyam Ouaissa, 2025). Each STRIDE category maps to particular vulnerabilities in smart-home implementations, for example, low-power sensors with weak authentication are susceptible to spoofing attacks, while resource-constrained devices may be disrupted by denial-of-service attacks at scale (Dobrilovic and Dragović, 2024; Nadifi et al., 2025; Neshenko et al., 2019).

In summary, smart-home security assurance must address heterogeneous devices, behavioural variability, limited observability, and the full threat spectrum as described by frameworks such as STRIDE. This motivates the discussion of profiling, anomaly detection, and intrusion detection mechanisms in the following sections.

2.3 Profiling in Security Assurance

Profiling is one of the fundamental analytical processes used in security assurance, enabling systematic characterisation of normal device and system behaviours in smart-home IoT settings (Sivanathan et al., 2019; Mudgerikar, Sharma, and Bertino, 2019). This section presents an organised discussion of profiling methods, both conventional and continuous. The essential ideas and aims of profiling are introduced first, followed by an examination of traditional paradigms including network-biased and host-biased approaches, with a critical look at their operational assumptions, methods, and limitations (Zarpelão et al., 2017; Bridges et al., 2020). Next, the evolution of profiling within Continuous Security Assurance (CSA) paradigms is examined, covering network-centric, host-centric, and rule-based continuous profiling (Farhan et al., 2025; Chen et al., 2023). Finally, comparative perspectives are drawn across paradigms, noting common deficits, single-modality bias, poor explainability, poor stealth attack coverage, and the absence of hybrid Sys-Net profiling, which inform the research directions adopted in this thesis (Lo et al., 2026; Sommer and Paxson, 2010).

2.3.1 Profiling in Traditional Security Assurance

Profiling Concepts and Objectives

Profiling constitutes a foundational mechanism in traditional security assurance, aimed at establishing behavioural baselines for devices and systems operating within smart-home environments. By systematically observing normal operational patterns, profiling enables the identification of deviations that may signal anomalous or malicious activity (Chandola, Banerjee, and Vipin Kumar, 2009; Anthi et al., 2019).

Behavioural profiles may be constructed across multiple observational planes. System-level profiling monitors internal device states, whereas network-level profiling captures communication behaviour and traffic dynamics across the smart-home ecosystem (Mudgerikar, Sharma, and Bertino, 2019; Sivanathan et al., 2019). These baselines provide comparative reference models that support early anomaly detection and situational awareness (Chandola, Banerjee, and Vipin Kumar, 2009).

To critically analyse the state of the art, recent profiling literature is categorised into two dominant paradigms:

- Network-biased profiling
- Host-biased profiling

Each category is examined in terms of technique, data source, operational assumptions, and limitations.

Network-Biased Profiling Approaches

Network-biased profiling focuses on modelling behavioural normality using traffic-centric telemetry such as packet statistics, flow dynamics, protocol usage, and connection patterns (Farhan et al., 2025; Ben Slimane, Abd-Elkawy, and Maqbool, 2024).

Review of Recent Network-Biased Profiling Studies Sarwar et al. (2023) investigated network-based anomaly detection in IoT environments using supervised machine-learning classifiers on network traffic data to identify malicious activity. While effective in modelling typical communication, the approach is limited by its inability to capture host-level compromise, such as on-device malware or privilege escalation, that does not produce discernible network anomalies (Hwang et al., 2023).

A. Verma and Ranga (2020) evaluated machine-learning classifiers for anomaly-based IoT intrusion detection, with a particular focus on denial-of-service attacks. They benchmarked classifiers on CIDDS-001, UNSW-NB15, and NSL-KDD and also evaluated classifier response time on Raspberry Pi hardware. The study therefore provides evidence for network-centric ML IDS performance and deployment cost, but it does not provide host-level behavioural telemetry.

Yaras and Dener (2024) proposed a hybrid 1D-CNN and LSTM intrusion-detection approach for analysing IoT network traffic and detecting malicious communication patterns. Although more effective at network-level monitoring by considering protocol semantics, the approach does not identify malicious behaviour inside devices , such as Living-off-the-Land (LotL) attacks, so host-level abuse remains unseen (Lo et al., 2026).

Koroniotis et al. (2019) introduced the BoT-IoT dataset, which combines legitimate and simulated IoT network traffic with multiple attack categories in a realistic testbed for network-forensic analytics. Because the dataset is network-traffic centric, it does not provide synchronised host-level telemetry for analysing attacks whose evidence remains primarily on the endpoint.

Analytical Observation Together, these network-biased profiling studies reveal high performance in identifying externally visible threats such as scanning, DDoS, and coordinated exfiltration attacks (Chaabouni et al., 2019). Nevertheless, a critical gap persists: by definition, they cannot detect attacks that occur solely at the host level, such as priv-

ilege escalation, execution of malicious scripts, or LotL malware (Lo et al., 2026; Bridges et al., 2020). Relying solely on network-biased profiling is therefore inadequate for ongoing security assurance, underscoring the necessity of hybrid profiling techniques that combine system-level and network-level telemetry (Lo et al., 2026).

Host-Biased Profiling Approaches

Host-biased profiling uses internal system telemetry, CPU utilisation, memory consumption, process activity, to define normal device operation. This approach is suited to detecting threats such as privilege escalation, malware code execution, and resource misuse, which may leave no noticeable network signatures (Mudgerikar, Sharma, and Bertino, 2019; Chen et al., 2023).

Review of Recent Host-Biased Profiling Studies Valluru et al. (2026) elaborated on a host-based anomaly detection framework using Random Forest classifiers trained on system call sequences. Although successful in detecting host-resident anomalies, the scheme cannot detect coordinated attacks perceivable only at the network level, constraining situational awareness in distributed IoT settings (Araya and Rifà-Pous, 2023; Jeyasoundari et al., 2026).

Pang et al. (2021) utilised deep learning, specifically autoencoder models, to learn latent host-level behaviour patterns. Although capable of detecting complex relationships, the learned representations are not very interpretable, making it difficult to attribute identified anomalies to specific attack goals (Samek, Montavon, Vedaldi, et al., 2019).

Narayanan and Vanitha (2025) examined blockchain-aided host profiling to confirm the reliability of device telemetry. Although distributed trust models improve data confidence, the computational and storage burdens of blockchain computing can limit practical deployment on resource-limited IoT systems (Canavese et al., 2024).

Analytical Observation Host-biased profiling is effective at identifying insider threats, malware, and privilege escalation which network-based approaches typically overlook (Chen et al., 2023; Kikissagbe and Adda, 2024). Its inherent weakness, however, is that it cannot observe coordinated network-level attacks or lateral device movements (Farhan et al., 2025). This security assurance gap motivates hybrid profiling strategies that combine host and network telemetry to ensure continuous, interpretable, and actionable security (Lo et al., 2026; Ardagna et al., 2017).

Comparative Limitations of Traditional Profiling

Traditional profiling methods suffer from systemic limitations:

1. **Static Baselines:** Fixed window-based profiles cannot support behavioural drifting (Gama et al., 2014; Pang et al., 2021; Aparcana-Tasayco, Deng, and Park, 2025).
2. **Single-Modality Observability:** Host-only or network-only visibility generates detection blind spots (Chen et al., 2023; Lo et al., 2026).
3. **Device Heterogeneity:** Incomparable IoT operating conditions reduce profiling generalisability (Vardakis et al., 2024).
4. **Resource Constraints:** Computational overhead limits deployability on smart-home devices with low power (Canavese et al., 2024; F. Hussain et al., 2020).

Taken together, these restrictions underscore the necessity of inter-layer, adaptive, and resource-conscious profiling schemes capable of enabling ongoing security assurance in dynamic smart-home environments (Ardagna et al., 2017; Watanabe et al., 2024).

2.3.2 Profiling in Continuous Security Assurance

This section reviews the development of profiling-based techniques within Continuous Security Assurance (CSA) for smart-home IoT settings. It begins with conceptual frameworks of continuous profiling, then critically surveys recent CSA literature partitioned into network-centric, host-centric, and rule-based profiling frameworks (Ardagna et al., 2017; Angermeir et al., 2024). Finally, literature gaps and shortcomings are identified, highlighting the necessity of hybrid, intent-aware, and longitudinal profiling solutions (Aparcana-Tasayco, Deng, and Park, 2025).

Continuous Profiling Concepts

Continuous profiling extends behavioural models into a temporally sustained security assurance construct. Rather than baselines obtained through fixed observation intervals, continuous profiling captures behavioural evolution through time, enabling adaptive and longitudinal anomaly observation (Gama et al., 2014; Watanabe et al., 2024).

In smart-home systems, device behaviour is inherently dynamic due to firmware updates, changes in workload, user interaction patterns, and environmental changes (Watanabe et al., 2024; Meidan et al., 2023). Continuous profiling consequently uses temporal modelling constructs, sliding observation windows, incremental learning, and drift-aware baselining, to maintain behavioural fidelity (Gama et al., 2014).

Important operational concepts include:

- **Dynamic baselining:** Behavioural norms constantly evolve to accept legitimate operational drift (Watanabe et al., 2024).
- **Sliding-window analysis:** Temporal segmentation identifies both transient and long-term deviations (Gama et al., 2014).
- **Longitudinal stability monitoring:** Persistent observation detects gradual se-

curity deterioration that point-in-time methods miss (Ardagna et al., 2017).

CSA-Oriented Profiling Literature Review

Current Continuous Security Assurance research demonstrates increasing interest in persistent behavioural monitoring of IoT and smart-home settings (Ardagna et al., 2017; Angermeir et al., 2024). To critically synthesise the literature, CSA profiling studies are analysed across three dimensions:

- Continuous monitoring capability
- Behavioural modelling strategy
- Scope of observation (network vs. host vs. hybrid)

Network-Centric Continuous Profiling

Farhan et al. (2025) proposed a deep-learning-driven continuous network intrusion detection framework that analyses IoT traffic flows in real time. While effective at capturing external threats, the approach is limited by its exclusive reliance on network data, lacking correlation with host-state information (Chen et al., 2023).

Ben Slimane et al. (2024) developed an adaptive network profiling method using flow statistics and protocol fingerprints to detect deviations in device communication patterns. The approach is limited to outgoing data and unable to detect insider misuse within the equipment itself (Chen et al., 2023).

Acar et al. (2020) explored privacy-aware traffic profiling of smart-home networks using encrypted traffic monitoring and fingerprinting methods. Encryption, however, prevents direct payload inspection and limits the semantic information available to traffic-analysis systems (Papadogiannaki and Ioannidis, 2021).

Together, these examinations show that network-focused persistent profiling assists scalable and non-intrusive monitoring, yet is by definition oblivious to internal device compromise and attacks that maintain network compliance (Mudgerikar, Sharma, and Bertino, 2019).

Host-Centric Continuous Profiling

Chen et al. (2023) introduced a host-based continuous monitoring framework leveraging system-call telemetry to model sequential patterns of device behaviour, enabling detection of anomalies arising from insider misuse or malware execution. While providing detailed operational visibility, the framework lacks awareness of network-level attack vectors (Farhan et al., 2025).

Kikissagbe and Adda (2024) proposed an ML-driven host profiling approach that utilises resource utilisation metrics for continuous anomaly detection. Although effective in detecting host-resident deviations, the method is sensitive to workload bursts and cannot generalise across multi-device interactions (Aparcana-Tasayco, Deng, and Park, 2025).

Araya and Rifà-Pous (2023) discuss adaptive learning approaches for addressing evolving device behaviour and concept drift in smart-home anomaly detection. Even with these developments, host-based solutions remain outside network context and thus lack the ability to identify coordinated or distributed attacks (Farhan et al., 2025).

Rule-Based and Threshold Continuous Profiling

Diana et al. (2025) reviewed adaptive threshold IDS architectures that continually score anomalies of normal behaviour, dynamically adjusting thresholds to changes in normal operation. Although computationally lightweight and interpretable, these methods are usually not context-aware and cannot reliably differentiate abnormal behaviour from

intentionally malicious actions (Sommer and Paxson, 2010).

Cross-Literature Analytical Gaps

Despite the progress made in CSA-based profiling, several systemic constraints persist:

1. Persistence of Single-Modality Bias: The majority of CSA profiling models are limited to either network or host telemetry, fragmenting behavioural visibility and preventing correlation of cyber–physical device states (Farhan et al., 2025; Chen et al., 2023).

2. Lack of Hybrid Behavioural Profiling: Although hybrid IDS architectures exist, true synchronised system–network behavioural profiling remains underexplored. Few studies jointly model internal resource behaviour and external communication patterns within a unified assurance baseline (Lo et al., 2026).

3. Absence of Intent-Aware Interpretation: CSA profiling literature predominantly focuses on anomaly detection rather than behavioural intent modelling. Adversarial goals, operational misuse, and threat progression are signalled statistically but rarely in a semantic way (Sommer and Paxson, 2010).

4. Limited Longitudinal Assurance Framing: Many "continuous" systems operate as streaming IDS pipelines rather than assurance-oriented behavioural monitors, limiting their applicability to long-term security posture evaluation (Ardagna et al., 2017; A. Shukla et al., 2022).

Limitations of CSA Profiling

The structural weakness of CSA profiling is its dependence on an initial benign baseline to estimate statistical parameters such as μ and σ (Chandola, Banerjee, and Vipin Kumar, 2009; Diana, Dini, and Paolini, 2025). Inaccurate or biased baseline windows can destabilise initial profiling fidelity, particularly in smart-home settings where the initial deployment period may already include stealthy adversarial activity (Aparcana-Tasayco, Deng, and Park, 2025).

In addition, profiling layers do not necessarily perform attack classification. They measure behavioural deviation without labelling it with semantic threat entities, positioning profiling as a descriptive monitoring construct rather than a decision engine (Sommer and Paxson, 2010; Chandola, Banerjee, and Vipin Kumar, 2009).

Therefore, traditional measures of classification accuracy are not suitable for evaluating profiling quality. Quality profiling is determined by:

- Statistical stability (Chandola, Banerjee, and Vipin Kumar, 2009)
- Behavioural interpretability (Samek, Montavon, Vedaldi, et al., 2019)
- Drift resilience (Watanabe et al., 2024)
- Suitability for downstream assurance analytics (Ardagna et al., 2017)

2.3.3 Comprehensive Analysis of Hybrid System–Network Behavioural Profiles

Table 2.2 presents a comparative analysis of recent profiling approaches across technique, data plane focus, and inherent limitations.

Table 2.2: Comparative Analysis of Profiling Approaches in Security Assurance

Paper	**Authors**	**Year**	**Technique**	**Data Plane**	**Key Limitation**
IoT Network Intrusion Detection using ML	Ferrag et al.	2023	Random Forest, Ensemble ML	Network	Limited host visibility; lacks system–network correlation
Smart-Home Traffic Privacy Profiling	Zhao et al.	2022	Deep Traffic Fingerprinting	Network	Encrypted traffic focus; no device state context
Adaptive Network Behaviour IDS	Ben Slimane et al.	2024	Deep Autoencoders	Network	Poor explainability; high computational cost
Host-based IDS for IoT	Chen et al.	2023	System-call ML Profiling	Host	Cannot detect network-borne stealth attacks
Collaborative HIDS Framework	Kikissagbe et al.	2024	Hybrid ML Classifiers	Host	Limited external traffic awareness
Statistical Smart-Home Profiling	Alshamrani et al.	2023	Probabilistic Modelling	Host	Static baseline; behavioural drift sensitivity
Rule-enhanced IoT IDS	Pang et al.	2021	Rule + ML Detection	Network	Signature dependence; weak zero-day coverage
BoT-IoT Dataset	Koroniotis et al.	2019	Labelled IoT network traffic dataset	Network	Network-centric; does not provide synchronised host telemetry (Koroniotis et al., 2019)
Continuous IoT Assurance Monitoring	Botto-Tobar et al.	2023	Sliding-window Profiling	Network	Network-only persistence; limited explainability
ML-driven CSA Framework	Sharma et al.	2022	Temporal Behaviour Modelling	Host	No hybrid telemetry fusion
IDS Performance Re-evaluation	Sommer and Paxson	2010	Behavioural IDS Analysis	Network	Limited stealth attack interpretability
Continuous Risk-aware Profiling	Behl et al.	2023	Risk-scored Behaviour Models	Hybrid	Conceptual hybrid only; lacks Sys–Net implementation

Cross-Study Observations

Lack of Hybrid Sys–Net Profiling: The majority of studies are limited to a single data plane. Network-based works rely on traffic metadata and flow statistics, while host-based systems rely on system calls, processes, and resource usage. The absence of system-network telemetry limits situational awareness and introduces detection blind spots (Bridges et al., 2020; Lo et al., 2026).

Limited Explainability: Advanced machine-learning and deep-learning algorithms enhance detection performance but tend to behave as black-box models. This reduces forensic interpretability and makes them less suitable for assurance-based security governance (Samek, Montavon, Vedaldi, et al., 2019; Neupane et al., 2022).

Poor Stealth Attack Coverage: Low-and-slow attacks that conceal malicious behaviour at both system and network levels are not easily detected. Network-only models miss host-level manipulation, while host-only models miss covert command-and-control activity (Chen et al., 2023; Lo et al., 2026).

Persistent Modality Bias in CSA: CSA research still demonstrates a biased use of profiling toward either host or network visibility. Hybrid real-time profiling based on temporal system behaviour and network context remains essentially under-explored (Lo et al., 2026).

Identified Gaps in Continuous Profiling Literature

Although significant progress has been made in network-based, host-based, and rule-based continuous profiling of smart-home IoT, the literature reveals several cumulative limitations pointing to a clear research gap. First, the majority of studies are limited to a single data plane, leading to poor visibility and detection blind spots where attacks altering both system and network states remain undetected (Farhan et al., 2025; Chen et al., 2023). Second, semantic interpretation of anomalous behaviour is generally ab-

sent; exceptions are reported statistics with no relationship between abnormal behaviour, threat development, or operational abuse (Sommer and Paxson, 2010). Third, many continuous profiling techniques assume constant or limited baselines, restricting adaptability to dynamic device behaviour, firmware changes, and evolving user interaction patterns (Watanabe et al., 2024; Aparcana-Tasayco, Deng, and Park, 2025). Finally, lightweight rule-based or threshold-based methods, while interpretable and computationally efficient, lack the context sensitivity needed to identify coordinated or stealthy attacks (Diana, Dini, and Paolini, 2025; Lo et al., 2026).

On the other hand, machine-learned IDS can learn complicated non-linear relationships in high-dimensional telemetry data, but often produce opaque decision models whose reasoning cannot be interpreted or verified by an analyst in practical contexts (Pang et al., 2021; Neupane et al., 2022; Samek, Montavon, Lapuschkin, et al., 2021).

All of these systemic constraints point to the necessity of a hybrid, cross-layer Sys-Net profiling system that combines host-level and network-level telemetry. Hybrid monitoring architectures allow richer behavioural modelling and improved detection of stealthy or low-and-slow attacks through correlation of internal device behaviour and external communication patterns (Lo et al., 2026).

2.4 Anomaly Detection in Security Assurance

Anomaly detection is an important analytical element of security assurance models, operating alongside behavioural profiling to detect aberrant behaviour of devices or systems. In smart-home IoT environments, where devices have dynamic workloads, receive firmware updates, and exhibit varied user behaviour, anomaly detection can provide early alerts to abnormal behaviour signifying misconfiguration, breach, or threats (Aparcana-Tasayco, Deng, and Park, 2025). This section provides an organised overview of anomaly detection methods, starting with traditional statistical and distance-based approaches,

followed by machine-learning and hybrid methods, and finally their integration into Continuous Security Assurance (CSA) paradigms.

2.4.1 Anomaly Detection in Traditional Security Assurance

Anomaly detection provides an additional analytical layer to behavioural profiling, vital for detecting deviations in the operational patterns of smart-home IoT devices (Lara et al., 2023; Pang et al., 2021).

Statistical and Distance-Based Detection

The predominant early anomaly detectors were based on statistical modelling of normal behaviour, in which probabilistic models are trained on historical data and anomalies outside normal ranges are flagged (Pang et al., 2021; Chandola, Banerjee, and Vipin Kumar, 2009). Distance-based techniques generalise this paradigm, measuring how dissimilar observed feature vectors are to reference clusters in multi-dimensional space (Chandola, Banerjee, and Vipin Kumar, 2009). These methods are appealing in smart-home IoT due to their relatively low computational requirements and conceptual simplicity, making them usable on resource-limited devices (Chandola, Banerjee, and Vipin Kumar, 2009; Pang et al., 2021).

Yet statistical and distance-based techniques are inherently predicated on the assumption that behavioural distributions do not change over time, an assumption frequently violated in dynamic environments when workloads change, firmware is updated, or legitimate users modify their behaviour patterns (Gama et al., 2014; Aparcana-Tasayco, Deng, and Park, 2025). As a result, these techniques tend to label benign operational variability as malicious deviation, producing high false-positive rates and low reliability in real-world use (Sommer and Paxson, 2010; Meidan et al., 2023).

Machine Learning–Based Detection

Machine learning introduced the ability to model complex and non-linear behavioural patterns. Supervised classifiers require labelled attack datasets to learn decision boundaries, whereas unsupervised techniques, such as clustering and autoencoders, infer anomalies without labelled data (Buczak and Guven, 2016; Chaabouni et al., 2019).

Unsupervised models are particularly prevalent in IoT environments due to the scarcity of labelled attack data (Pang et al., 2021). Explainable AI techniques have been explored to enhance transparency in anomaly decisions (Samek, Montavon, Lapuschkin, et al., 2021). Despite improved detection capability, these methods often introduce computational overhead and limited interpretability (Pang et al., 2021; Neupane et al., 2022).

Data-Plane Bias in Traditional Detection

A noticeable limitation in traditional anomaly detection research for IoT environments is a structural bias toward single-plane observability, where techniques are developed and evaluated using either network-centric or host-centric data, but rarely both in an integrated manner (Lo et al., 2026). Several network-centric detection studies characterise abnormal behaviour using traffic features such as packet rates, flow durations, and protocol usage (Ring et al., 2019). By focusing solely on network traffic, these methods are unable to detect host-residing anomalies that do not generate prominent network signatures (Farhan et al., 2025).

Conversely, host-centric detection approaches analyse device-level metrics such as CPU usage, system calls, and process activity (Bridges et al., 2020; Mudgerikar, Sharma, and Bertino, 2019; Pang et al., 2021). Although providing significant insights into on-device activity, they give no insight into organised attack patterns in a network (Chen et al., 2023).

Such single-plane biases limit the capabilities of holistic anomaly detection since

every observational domain measures only part of the behavioural signals (Chen et al., 2023). This separation reduces the capacity to map cross-domain indicators of complex threats within smart-home settings, motivating the development of hybrid Sys-Net anomaly detection schemes (Lo et al., 2026).

Limitations of Traditional Anomaly Detection

Traditional anomaly detection mechanisms have several systemic constraints restricting their usage in assurance-driven smart-home environments (Aparcana-Tasayco, Deng, and Park, 2025). These methods frequently generate high false positives due to static thresholds and rigid behavioural models that cannot capture legitimate operational variation across devices (Meidan et al., 2023). They also lack contextual awareness, examining features independently without considering cross-device connections or incorporating understanding of device state and activity patterns (Pang et al., 2021). Machine learning detectors are often not interpretable, models such as deep autoencoders, LSTMs, or clustering algorithms can be accurate but opaque (Samek, Montavon, Lapuschkin, et al., 2021). Finally, Living-off-the-Land (LotL) host misuse attacks, low-volume exfiltration, or slow-moving malware bring negligible variations in normal monitoring values, inherently evading methods that need statistically significant or substantial feature differences (Lo et al., 2026).

2.4.2 Anomaly Detection in Continuous Security Assurance

Continuous Security Assurance (CSA) extends anomaly detection from static deviation identification to temporally persistent behavioural monitoring (Ardagna et al., 2017; Watanabe et al., 2024).

Purpose of Anomaly Detection in Continuous Security Assurance

Within the proposed CSA architecture, anomaly detection serves three assurance functions:

- Quantification of behavioural deviation over time (Chandola, Banerjee, and Vipin Kumar, 2009).
- Identification of statistically significant anomalies beyond natural profiling variance (Aparcana-Tasayco, Deng, and Park, 2025).
- Escalation signalling for downstream IDS and STRIDE-based semantic analysis (Shevchenko et al., 2018).

Unlike binary IDS outputs, anomaly detection produces continuous deviation scores that enable graded security assessment and behavioural risk interpretation (Chandola, Banerjee, and Vipin Kumar, 2009).

Behavioural Modelling Techniques in CSA

CSA anomaly detection methods have evolved to address the dynamic and heterogeneous nature of smart-home IoT environments and can be categorised into four dominant methodological groups:

Statistical Deviation Models: Statistical deviation techniques model normal behaviour by estimating feature distributions and defining thresholds, such as $\mu \pm k\sigma$, to identify outliers (Chandola, Banerjee, and Vipin Kumar, 2009). These methods are computationally lightweight, interpretable, and suitable for resource-constrained devices (Chandola, Banerjee, and Vipin Kumar, 2009). Nevertheless, they use fixed distribution assumptions, making them vulnerable to acceptable behavioural drift (Aparcana-Tasayco, Deng, and Park, 2025).

Machine Learning Approaches: Machine learning models, such as supervised (SVM, Random Forest) and unsupervised (clustering, PCA, autoencoders) methods, are trained on historical telemetry (Buczak and Guven, 2016; Géron, 2022). They are capable of finding non-linear relationships and small anomalies, but are sensitive to concept drift and require well-represented labelled datasets (Andresini et al., 2021).

Hybrid and Deep Learning Methods: Deep learning methods including LSTM, CNN, or integrative statistical-ML pipelines take advantage of both temporal and spatial correlations in telemetry data (Pang et al., 2021). These promise performance improvements but are computationally intensive and lack explainability (Samek, Montavon, Vedaldi, et al., 2019; Neupane et al., 2022).

Adaptive Detection Systems: Adaptive systems can update anomaly-detection models as attack patterns evolve (Watanabe et al., 2024). Watanabe et al. continuously capture IoT traffic, adaptively generate an anomaly-detection model, and dynamically update it to reflect changes in anomalous traffic patterns. This demonstrates real-time adaptation to evolving attacks, although the approach remains network-side.

Persistence of Plane Bias in CSA Detection

CSA anomaly detection, despite architectural developments, still suffers structural limitations:

- Constant use of network-only telemetry (Farhan et al., 2025; Ben Slimane, Abd-Elkawy, and Maqbool, 2024).
- Host behavioural analysis isolated from network context (Chen et al., 2023; Kikissagbe and Adda, 2024).
- Low cross-layer correlation (Lo et al., 2026).

The hybrid Sys-Net anomaly scoring literature on continuous assurance remains minimal (Lo et al., 2026).

Limitations of CSA Anomaly Detection

Recent analyses identify the following permanent limitations:

- Inadequate behavioural baseline articulation (Aparcana-Tasayco, Deng, and Park, 2025).
- Elevated false positives from benign variability (Meidan et al., 2023).
- Poor generalisation to previously unseen conditions (Sommer and Paxson, 2010).
- High computational overhead in deep models (F. Hussain et al., 2020).
- Limited interpretability of detection outputs (Neupane et al., 2022).
- Strong reliance on labelled datasets (Andresini et al., 2021).
- Sensitivity to minor deviations (Sommer and Paxson, 2010).
- Inability to address long-term behavioural drift (Aparcana-Tasayco, Deng, and Park, 2025).
- Limited scalability in heterogeneous deployments (Vardakis et al., 2024).
- Dataset realism gaps (Engelen, Rimmer, and Joosen, 2021; Ring et al., 2019).

Absence of Intent-Aware Interpretation

A critical gap in CSA anomaly detection is the absence of semantic interpretation. Detection outputs quantify deviation but do not explain adversarial intent or threat category (Sommer and Paxson, 2010). As a result, anomaly signals require downstream analytical layers for operational meaning (Sommer and Paxson, 2010).

2.4.3 Comprehensive Analysis of Anomaly Detection Models

Table 2.3 presents a comparative review of representative anomaly detection studies in smart-home and IoT security assurance literature.

Table 2.3: Analysis of Anomaly Detection Approaches in Smart-Home Security

Paper	**Authors**	**Year**	**Technique**	**Data Plane**	**Key Limitation**
Deep Anomaly Detection for IoT	Pang et al.	2021	Deep Autoencoder	Network	High computational overhead; limited explainability; no host correlation
ML-Based IDS Survey	Saleem et al.	2022	Supervised / Unsupervised ML	Hybrid (conceptual)	Operational deployments remain single-plane; lacks behavioural grounding
IoT IDS Survey	Chaabouni et al.	2019	ML Classification	Network	Poor contextual awareness; limited stealth attack visibility
Explainable AI methods	Samek et al.	2019	XAI methods	General	Improves transparency and interpretability of model decisions (Samek, Montavon, Vedaldi, et al., 2019)
IoT Security Monitoring Framework	Botto Tobar et al.	2023	Streaming Statistical Detection	Network	Limited cross-layer behavioural modelling
ML-Based IoT Security Framework	Sharma et al.	2022	Hybrid ML Detection	Hybrid (logical)	Integration lacks real-time Sys–Net synchronisation
Statistical IoT Detection Survey	Adhikari et al.	2024	Statistical Thresholding	Network	Sensitive to benign variability; weak stealth detection
IDS Modelling Techniques Review	Rajaon	2025	SVM / RF / Clustering	Network	Label dependency; behavioural drift sensitivity
Deep Learning IoT Detection	Zhang et al.	2023	CNN / LSTM	Network	High resource overhead; poor deployment feasibility
Adaptive IoT Detection Systems	Watanabe	2024	Self-Learning Detection	Network	No explicit behavioural baseline; weak interpretability
Smart-Home IDS Analysis	Sarwar et al.	2023	ML-Based IDS Pipeline	Network	Anomaly detection implicit; lacks behavioural visibility
IoT Detection Survey	Aparcana-Tasayco	2025	Hybrid Detection Survey	Network	No assurance-oriented anomaly scoring paradigm

The comparative evaluation reveals that existing anomaly detection approaches remain constrained by three systemic limitations:

- Persistent reliance on single-plane telemetry (network or host) (Lo et al., 2026).
- Limited explainability of anomaly decisions (Samek, Montavon, Lapuschkin, et al., 2021).
- Inadequate detection of stealth-oriented or low-signal attacks (Lo et al., 2026).

These limitations highlight the absence of profiling-driven hybrid anomaly detection capable of producing behaviourally interpretable deviation scores, motivating the contribution proposed in this work.

2.4.4 Comparative Analysis of Black-Box Machine Learning Techniques in IoT Anomaly Detection

A further dimension of analysis concerns the interpretability and operational transparency of the machine learning models deployed for anomaly detection and intrusion classification in smart-home IoT environments. The dominant pattern in current literature is the use of black-box models, methods whose internal decision processes are not directly accessible or interpretable to security operators, which creates significant challenges for trust, auditability, and operational integration within assurance-driven architectures (Samek, Montavon, Vedaldi, et al., 2019; Neupane et al., 2022).

Table 2.4 summarises representative black-box and explainability-enhanced machine learning approaches evaluated in the context of IoT anomaly detection and IDS.

Table 2.4: Comparative Analysis of Black-Box and Explainability-Oriented Machine Learning Techniques in IoT Anomaly Detection

Paper	**Authors**	**Year**	**ML Technique**	**Interpretability**	**Relevance / Limitation for CSA**
Deep Learning Anomaly Detection	Pang et al.	2021	Deep Autoencoder	Black-box	High detection accuracy on benchmark datasets; internal decision logic opaque; incompatible with assurance-based audit requirements (Pang et al., 2021)
CNN/LSTM IoT Detection	Yaras and Dener	2024	CNN / LSTM	Black-box	Strong temporal anomaly capture; poor explainability; high computational cost limits gateway deployment (Yaras and Dener, 2024)
Explainable AI methods	Samek et al.	2019	Attribution and explanation methods	Interpretable	XAI methods improve model transparency, while the reliability of explanations remains an evaluation concern (Samek, Montavon, Vedaldi, et al., 2019)
Ensemble ML IoT IDS	Sarwar et al.	2023	Random Forest (Ensemble)	Partial (feature importance)	Built-in feature importance provides partial interpretability; competitive with deep learning while remaining deployable (Sarwar et al., 2023; M. Z. Hussain et al., 2026)

Continued on next page

Paper	**Authors**	**Year**	**ML Technique**	**Interpretability**	**Relevance / Limitation for CSA**
Adversarial ML in IDS	Winanto and Idris	2026	Deep Neural Network	Black-box	Susceptible to adversarial perturbations; black-box classifiers fail to detect evasion attempts that satisfy decision boundary but violate physical consistency constraints (Winanto and Idris, 2026; Alajaji et al., 2026)
Adaptive Self-Learning IDS	Watanabe	2024	Self-Learning / Online ML	Semi-interpretable	Improves drift tolerance but lacks explicit behavioural baseline; anomaly reasoning not auditable (Watanabe et al., 2024)
Smart-Home ML IDS Evaluation	Dinesh et al.	2026	Ensemble ML	Partial	Heterogeneous device generalisation constrained; no STRIDE-aligned semantic output (Dinesh et al., 2026)
Explainable IDS Pipeline	Smiliotopoulos et al.	2026	DNN + XAI overlay	Partially interpretable	XAI overlay added post-hoc; core model remains black-box; evasion-resilience not evaluated (Smiliotopoulos et al., 2026)
Host Anomaly via RF	Valluru et al.	2026	Random Forest	Interpretable	Interpretable and stable; limited to host-plane; no network correlation (Valluru, S. Shukla, and Pateriya, 2026)

Continued on next page

Paper	**Authors**	**Year**	**ML Technique**	**Interpretability**	**Relevance / Limitation for CSA**
Lifecycle Security Anomaly Detection	Ron et al.	2025	Multi-layer ML / Federated anomaly detection	Explainable (XAI)	Integrates device- and network-level monitoring with cross-layer anomaly correlation; does not provide STRIDE-aligned semantic interpretation, continuous security-posture quantification, or explicit adversarial-robustness evaluation (Ron et al., 2025)
Adversarial Robustness in IDS	Alhajjar et al.	2020	Multiple (black-box)	Black-box	Demonstrates that black-box classifiers achieve high evasion rates even without model access; underlines need for physical consistency defence layers (Alhajjar, Maxwell, and Bastian, 2021)

The analysis of black-box and explainability-oriented techniques reveals several patterns that are directly relevant to the design of the proposed CSA framework. First, black-box models, including deep autoencoders, CNN/LSTM architectures, and deep neural networks, consistently achieve the highest anomaly detection rates on benchmark evaluations but remain fundamentally opaque, preventing the semantic attribution of detected anomalies to structured threat categories (Pang et al., 2021; Neupane et al., 2022). This opacity is not merely an academic concern; in assurance-driven operational settings, security operators require not only a detection signal but an interpretable rationale that supports decision-making and accountability (Samek, Montavon, Vedaldi, et al., 2019; Neupane et al., 2022).

Second, post-hoc explainability overlays, such as SHAP and LIME applied to deep models, partially address interpretability but do not alter the underlying decision boundary of the model, meaning that adversarial perturbations can still evade detection while superficially satisfying the explanation interface (Slack et al., 2020). This limitation is particularly acute in adversarial settings, where an informed attacker can exploit the gap between the model's learned decision boundary and the physical constraints of the real system (Biggio and Roli, 2018; Vassilev et al., 2025).

Third, ensemble methods, particularly Random Forest, offer a practically viable balance between detection performance, computational efficiency, and partial interpretability through built-in feature importance measures (Breiman, 2001; Buczak and Guven, 2016). These properties align with the operational requirements of CSA frameworks, where models must be deployed continuously on resource-constrained smart-home gateways and must produce outputs that are auditable and semantically meaningful (Ardagna et al., 2017; F. Hussain et al., 2020).

The overall picture emerging from this analysis is that neither pure black-box nor fully transparent models currently satisfy all requirements for CSA-oriented anomaly detection and intrusion classification in smart-home environments (Neupane et al., 2022). The proposed framework addresses this tension by adopting a profiling-grounded anomaly

detection layer that produces statistically interpretable deviation scores, retaining explainability, and a Random Forest–based IDS that provides semantic STRIDE-aligned classification , retaining partial transparency, while adding a physically constrained consistency verification layer as an adversarial defence that operates independently of the classifier's decision boundary (Goodfellow, Shlens, and Szegedy, 2015; Vassilev et al., 2025).

2.5 Intrusion Detection Systems (IDS) in Security Assurance

2.5.1 IDS in Traditional Security Assurance

Intrusion Detection Systems (IDS) are critical components of traditional security assurance, providing automated mechanisms to identify potential attacks by analysing system and network behaviours (Khraisat, Gondal, Vamplew, and Kamruzzaman, 2019). IDS methods typically classify into:

- **Signature-Based IDS:** Detects known attack patterns using pre-defined signatures (Khraisat, Gondal, Vamplew, and Kamruzzaman, 2019).
- **Anomaly-Based IDS:** Flags deviations from established baselines, enabling detection of previously unseen attacks (Khraisat, Gondal, Vamplew, and Kamruzzaman, 2019).
- **Hybrid IDS:** Combines signature and anomaly-based approaches to enhance coverage and accuracy (Khraisat, Gondal, Vamplew, Kamruzzaman, and Alazab, 2019).

Deployment in Smart-Home Environments

- **Network-Based IDS (NIDS):** Monitors inter-device traffic for suspicious patterns without impacting device operation (Diana, Dini, and Paolini, 2025; Farhan et al., 2025; Ben Slimane, Abd-Elkawy, and Maqbool, 2024).
- **Host-Based IDS (HIDS):** Monitors device-level events, logs, and system calls for possible intrusions (Pang et al., 2021; Chen et al., 2023; Kikissagbe and Adda, 2024; Bridges et al., 2020).

Limitations of IDS in Traditional Security Assurance

Traditional IDS approaches in smart-home settings have two key limitations. Alerts are usually poorly contextualised semantically or behaviourally validated (Sommer and Paxson, 2010): an IDS may identify anomalies but rarely interprets their operational meaning, attack intent, severity, or appropriate mitigation steps in a continuous assurance context (Sommer and Paxson, 2010). Additionally, raw traffic classification can generate high false-positive rates (Ahmad et al., 2021), particularly when small legitimate deviations , firmware updates, transient workload spikes, are misclassified as malicious without contextual modelling or host-level correlation (Meidan et al., 2023).

2.5.2 IDS in Continuous Security Assurance

Under CSA, IDS is viewed not as a standalone decision engine but as part of a wider security pipeline operating across multiple layers (Imran et al., 2026). Key principles include:

- **Behaviour-validated invocation:** Profiling and anomaly detection confirm statistically significant deviation before IDS classification is triggered.

- **Integration with CSA pipeline:** Profiling, anomaly detection, and IDS work as sequential assurance layers rather than isolated components.
- **STRIDE-aligned interpretation:** Identified anomalies are mapped to organised adversarial labels (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) enabling explainable and prioritisable notifications (Mariya Ouaissa and Mariyam Ouaissa, 2025).

Limitations of IDS in CSA

Despite the development of sophisticated ML-based IDS systems, their application within CSA for smart-home environments faces significant challenges:

- **Continuous classification without behavioural validation:** Shafiq et al. (2021) proposed a machine learning IDS for IoT traffic streaming. Despite effectiveness in detecting statistical anomalies, the system does not contextualise deviations behaviourally, producing alerts not necessarily linked to true security threats.
- **High false-positive rates under benign variability:** Araya and Rifà-Pous (2023) review anomaly-based detection approaches for continuous monitoring in smart-home environments. However, benign but significant variations, such as automatic firmware updates or transient workload spikes , are often perceived as malicious, increasing false-positive rates and administrative load.
- **Limited interpretability of deep learning-based IDS:** Yaras and Dener (2024) and Smiliotopoulos et al. (2026) performed anomaly detection using CNNs and LSTMs. These models achieve high detection accuracy but provide limited explanatory information for generated alerts.
- **Poor robustness to heterogeneous smart-home devices:** Dinesh et al. (2026) present an IDS trained on benchmark device traffic with restricted generalisation

across heterogeneous device types showing different communication patterns.

- **Dataset overfitting to attack-centric benchmarks:** Shafiq et al. (2021) tested the IDS on datasets with high percentages of attack instances, resulting in models optimised for benchmark conditions at the cost of real-world fidelity (Koroniotis et al., 2019; Engelen, Rimmer, and Joosen, 2021).

- **Weak mapping between alerts and adversarial intent:** Shukla et al. (2022) observe that existing security-assurance methods can flag anomalies but are less effective at defining their impact and potential risks.

- **Limited visibility into host-level resource abuse:** Mohamed Amine Ferrag et al. (2020) focus primarily on network traffic features, leaving host-level anomalies such as CPU or memory exploitation largely undetected.

- **High computational overhead for real-time deployment:** Yaras and Dener (2024) employ complex ML models, making real-time execution on resource-constrained smart-home hubs challenging (Alfahaid et al., 2025).

- **Lack of integration with assurance pipelines:** Recent work has moved towards integrated multi-layer monitoring. Ron et al. (2025), for example, combine device-level and network-level anomaly indicators within a common security pipeline and automate their cross-layer correlation. However, the framework does not extend this integration to STRIDE-aligned semantic threat interpretation, continuous security-posture quantification, or explicit adversarial-robustness verification.

- **Absence of STRIDE-aligned threat interpretation:** Watanabe et al. (2024) introduces adaptive detection systems without semantic alignment of anomalies with STRIDE threat categories, limiting interpretability and downstream decision-making (Shevchenko et al., 2018).

2.5.3 Comprehensive Analysis of IDS and STRIDE Approaches

Khraisat et al. (2019) present a systematic survey of intrusion detection techniques, distinguishing between signature-based and anomaly-based methods (Khraisat, Gondal, Vamplew, and Kamruzzaman, 2019). While their taxonomy provides a foundation for selecting a particular IDS approach, this is a survey rather than an experimentally proven and validated continuous assurance architecture for coordinating device-specific profiling, anomaly escalation, and semantic classification. Pendleton et al. (2016), through their analysis of systems security metrics and the challenges of measuring security states, investigate the principles of defining metrics for security risk management (Pendleton et al., 2016). Their work contributes to security posture measurement; however, it does not provide an empirical evaluation of a signature-based IDS or runtime intrusion classification based on STRIDE. In their review, Chaabouni et al. (2019) discuss learning-based network intrusion detection methods for IoT, focusing on a set of statistical and machine-learning techniques capable of increasing the effectiveness of network-based monitoring in IoT environments. They also provide an overview of the challenges associated with these methods (Chaabouni et al., 2019). Pang et al. (2021) provide an overview of deep anomaly-detection methods, establishing a conceptual framework for understanding representation learning and anomaly scoring (Pang et al., 2021). This review, however, does not automatically confirm an end-to-end smart-home CSA pipeline or provide an example of understandable and interpretable STRIDE threat attribution using deep anomaly detection alone.

Other studies illustrate varying degrees of integration in the relationship between detection, modelling, and operational security interpretation. The proposed monitoring architecture and the degree to which behavioural evidence is used for intrusion detection are considered in the context of the study by Botto Tobar et al. (2023) (Botto-Tobar, Rehan, and R. P. Verma, 2023). If profiling, anomaly escalation, and assurance assessment are part of a single integrated workflow, the study is relevant to the present research; if they are implemented separately, this distinction must be acknowledged. Sharma et

al. (2022), presented in Table 2.5, demonstrate the application of conventional machine-learning (ML) classifiers used in IoT intrusion detection. In the present study, the distinction concerns whether behavioural patterns are validated for individual devices before a classification decision is made and whether the resulting evidence is used to assess security posture continuously. Shevchenko et al. (2018) describe available threat-modelling techniques, such as STRIDE, for structured threat identification and architectural reasoning (Shevchenko et al., 2018). However, it is important to note that STRIDE is a threat taxonomy rather than an experimentally verified runtime detector and should therefore not be confused with the statistical methods needed to detect malicious behaviour. The study cited by Ferrag et al. (2023) is considered for its role in machine-learning-based threat detection and interpretation. However, additional evidence is required to establish its contribution to cross-layer profiling, behavioural validation, and security-posture integration.

Al-Hawawreh et al. (2024) are included in the comparison of machine-learning-based intrusion-detection approaches evaluated using available security datasets. A key question for such approaches is whether the reported performance extends to datasets and attack distributions other than those employed during the system evaluation, particularly when host and network observations are assessed separately. In the context of training data, attack labels, and machine-learning-based detection of IoT attacks, Shafiq et al. (2021) are directly relevant to classification performance (Shafiq et al., 2021). However, classification performance alone does not support the claim that an alert has been behaviourally validated against the normal operating profile of a specific device. The significance of Nguyen et al. (2025) is considered in the context of adaptive intrusion detection, particularly whether adaptation can support explicit semantic attribution and continuous assurance decisions. Finally, Imran et al. (2025) conceptually define and discuss the relationship between continuous monitoring and security assurance (Imran et al., 2026). The present thesis builds upon this line of research by implementing and evaluating a physically constrained verification layer, classification based on known behavioural patterns, and continuous hybrid behavioural profiling. Consequently, although

these studies may share certain similarities, they must be considered separately because they implement and evaluate different intended functions and purposes rather than being treated as directly comparable IDS models.

The reviewed literature illustrates that intrusion detection accuracy, threat modelling, and continuous security assurance are interrelated but distinct research topics. Surveys define taxonomies of methods, classifier studies illustrate the performance of classifiers using specific datasets, threat modelling concerns understanding and reasoning about adversarial goals or intentions, and assurance-oriented architectures aim to maintain an operationally useful and interpretable assessment of security. This thesis addresses the unresolved challenge of their integration: combining synchronised evidence at both host and network levels, distinguishing meaningful deviations from variability that naturally occurs in the absence of adversarial activity, assigning supported detections to threat categories, and maintaining a security posture measure under adversarial manipulation. The methodological contributions and corresponding limitations presented in Table 2.5 explain the contribution and limitations of each reviewed work in relation to this integrated requirement. The identified limitations indicate the scope of the reviewed contributions and do not suggest that all studies were intended to solve the complete CSA problem.

Table 2.5: Comparative Analysis of IDS and STRIDE-Oriented Approaches

Paper	Authors	Year	Classifier / Framework	Invocation Mode	Key Limitation
IDS Taxonomy	Khraisat et al.	2019	IDS methods and dataset survey	Survey; no single runtime pipeline	Does not empirically evaluate a continuous hybrid CSA architecture
Security Metrics	Pendleton et al.	2016	Systems security metrics survey	Not an IDS evaluation	Addresses security measurement, not runtime attack classification or STRIDE mapping
IoT IDS Survey	Chaabouni et al.	2019	Learning-based IoT NIDS survey	Survey; no single runtime classifier	Does not evaluate a unified host-network smart-home CSA pipeline
Deep Anomaly Detection Review	Pang et al.	2021	Deep anomaly-detection review	Review; not a single IDS implementation	Does not itself evaluate a smart-home CSA pipeline or STRIDE attribution
Smart-Home Security Guidance	Botto-Tobar, Rehan, and R. P. Verma	2023	Homeowner protection strategies	Not a runtime detector	Provides security guidance; not an empirically evaluated CSA monitoring architecture
IoT Security Survey	Sharma et al.	2022	Random Forest / SVM	Continuous	Classification not behaviour-validated
Threat-Modelling Methods	Shevchenko et al.	2018	Summary of threat-modelling methods, including STRIDE	Design-time threat modelling	Describes threat modelling; it is not a runtime telemetry detector
Explainable IDS	Mohamed A. Ferrag et al.	2023	Random Forest IDS	Continuous	Explainable but not assurance-integrated
Smart-Home IDS Evaluation	Al-Hawawreh et al.	2024	Ensemble ML IDS	Continuous	Dataset-centric; lacks semantic threat mapping
CorrAUC Bot-IoT Traffic Detection	Shafiq et al.	2021	Machine-learning traffic detection	Dataset-based evaluation	Does not establish device-specific hybrid host-network assurance
Adaptive IDS Systems	Nguyen et al.	2025	Adaptive IDS	Continuous	Weak mapping between alerts and intent
CSA and Live Security Posture	Imran et al.	2026	Host/network monitoring, STRIDE, and posture framework	Proposed framework and case study	Case-study evidence is not a protocol-matched evaluation of the present experiments

The comparison in Table 2.5 summarises complementary contributions rather than a direct ranking of equivalent IDS models. It also shows why the proposed design is assessed as a specific combination of device-specific profiling, anomaly-triggered STRIDE classification, and physical-consistency verification.

2.6 Synthesis: Limitations of Traditional Security Assurance

The reviewed approaches expose recurring limitations in smart-home IoT security assurance: profiling, anomaly detection, and IDS are often treated separately; monitoring can remain event-driven and reactive; periodic evaluation and static baselines do not readily accommodate behavioural drift; and heterogeneous device characteristics complicate the application of traditional security products. As discussed in the detailed analysis above, these limitations motivate continuous security assurance integrating profiling, anomaly detection, and IDS into a unified system for ongoing monitoring (A. Shukla et al., 2022; Pang et al., 2021; Ardagna et al., 2017; Gama et al., 2014; Aparcana-Tasayco, Deng, and Park, 2025; F. Hussain et al., 2020; Vardakis et al., 2024; Imran et al., 2026).

2.7 Rationale for Continuous Security Assurance (CSA)

The emergence of Continuous Security Assurance is informed by several critical constraints inherent in traditional security approaches in smart-home IoT environments (Ardagna et al., 2017; Imran et al., 2026):

- **Behavioural drift in IoT devices:** Smart-home devices exhibit substantial behavioural change, rendering instantiated baselines and regular profiling ineffective

at detecting threats accurately (Watanabe et al., 2024; Meidan et al., 2023).

- **Stealthy and low-and-slow attacks:** Sophisticated distributed risks and subtle adversarial behaviours are difficult to notice under intermittent monitoring (Lo et al., 2026).
- **Limitations of threshold-based models:** Traditional anomaly detection and IDS systems based on predetermined thresholds produce high false-positive rates and ignore the dynamic behaviour of devices (Pang et al., 2021; Meidan et al., 2023).
- **Continuous monitoring principles:** CSA emphasises real-time runtime assurance, incorporating temporal context, streaming analytics, and adaptive thresholds (Ardagna et al., 2017; Angermeir et al., 2024).
- **Foundations in recent literature:** Empirical evidence demonstrates that a continuous feedback loop with profiling, anomaly detection, and IDS boosts detection and strengthens resistance to new threats (Watanabe et al., 2024; Andresini et al., 2021).

2.8 Adversarial Robustness in IoT Intrusion Detection

Statistical learning models derive decision rules from the distributional characteristics of training data, rendering them susceptible to deliberate input manipulation (Goodfellow, Shlens, and Szegedy, 2015; Biggio and Roli, 2018). An adversary exploiting this property introduces bounded perturbations to feature vectors, shifting the input across the classifier boundary while preserving its surface-level plausibility, so that a malicious payload escapes detection without triggering conventional inspection (Papernot et al., 2017).

In network intrusion detection specifically, this vulnerability translates into a concrete operational threat: attack traffic is reclassified as benign, allowing hostile activity

to propagate through the monitored environment without generating an alert (Alhajjar, Maxwell, and Bastian, 2021; Vassilev et al., 2025). The threat is not confined to white-box scenarios; black-box strategies that rely solely on querying the classifier and observing predicted labels have been shown to achieve comparable evasion rates (Papernot et al., 2017; Biggio and Roli, 2018).

A growing body of research has demonstrated that machine learning-based IDS, despite strong benchmark performance, remain fundamentally vulnerable to adversarial perturbations, deliberately crafted feature-space modifications designed to cause misclassification while preserving semantic plausibility (Apruzzese et al., 2022; Rosenberg et al., 2021). In network intrusion detection, adversarial attacks apply bounded perturbations within a defined epsilon budget, exploiting decision boundary geometry without necessarily altering the observable physical behaviour of the attack (Biggio and Roli, 2018).

Two attacker models are commonly distinguished. An oblivious attacker optimises evasion without knowledge of any deployed defence layer (Carlini and Wagner, 2017). An adaptive attacker incorporates explicit knowledge of the defence mechanism into the attack objective, representing a significantly stronger and more realistic threat model for deployed security systems (Carlini and Wagner, 2017; Tramèr et al., 2020).

Adversarial training, augmenting the training dataset with adversarially perturbed instances, has been proposed as a primary defence strategy, and has been shown to reduce attack success rates substantially (Madry et al., 2017). However, adversarial training alone does not eliminate evasion vulnerability, particularly against adaptive attackers aware of the augmentation procedure (Tramèr et al., 2020). This has motivated research into complementary defence layers operating independently of the classifier, using physical consistency constraints or statistical consistency checks to detect perturbations that satisfy the classifier's decision boundary but violate expected operational behaviour (Vassilev et al., 2025).

In the IoT and smart-home security domain specifically, adversarial robustness

remains substantially underexplored. Most published IDS evaluations assess classification performance under clean test conditions without considering an adversary who actively attempts evasion (Apruzzese et al., 2022). This gap is particularly significant in the context of Continuous Security Assurance, where sustained detection reliability under adversarial pressure is a foundational requirement (Rosenberg et al., 2021). The present work addresses this gap directly by evaluating the proposed CSA framework against both oblivious and adaptive attackers, and by introducing a physically constrained consistency verification layer as a defence that remains effective even when the underlying classifier is successfully evaded.

2.9 Key Research Challenges

The literature highlights a major limitation of existing smart-home security monitoring methods: inadequate support for cross-layer observation of network and host behaviour, including limited representation of meaningful relationships between the two telemetry planes and the heterogeneous nature of smart-home device behaviour. The resulting research challenge is to design a synchronised hybrid feature representation that preserves meaningful relationships between the two telemetry planes and accommodates the operational heterogeneity of smart-home devices (Khaleghi et al., 2013; Ardagna et al., 2017; A. Shukla et al., 2022).

The problem arises when different device classes display different operating characteristics, as fixed thresholds and pooled behavioural baselines may not be sufficient to distinguish normal variability from security-relevant deviations. A binary alert alone is not sufficient for continuous assurance; a security-posture measure should remain meaningful under benign behavioural drift and genuine adversarial perturbation. In smart-home environments, where device workloads are irregular and the boundary between normal variability and early-stage compromise may be subtle, this stability requirement is especially demanding (Chandola, Banerjee, and Vipin Kumar, 2009; Aparcana-Tasayco,

Deng, and Park, 2025; Watanabe et al., 2024; Meidan et al., 2023).

Anomaly scores or intrusion alerts provide only partial indications of a system's security state. A useful continuous assurance architecture should link measured behaviour with the security state and supported threat interpretations, while distinguishing anomaly classification from threat modelling and risk assessment. The challenge is to transform detection outputs from different layers into understandable threat categories and continuously measure the security condition of the system without treating a single alert as a complete representation of that condition (Sommer and Paxson, 2010; Shevchenko et al., 2018; Pendleton et al., 2016).

Evaluating classification performance on ordinary held-out data is not sufficient to establish the generalisability and security boundaries of the proposed assurance approach. Learning-based intrusion-detection systems may remain vulnerable when adversaries deliberately manipulate observable features. Therefore, adversarial reliability must be evaluated under explicitly defined threat models, while verification mechanisms consider relevant physical or statistical consistency constraints. These evaluations are necessary to establish the operational boundaries and generalisation of the proposed assurance approach (Goodfellow, Shlens, and Szegedy, 2015; Biggio and Roli, 2018; Papernot et al., 2017; Alhajjar, Maxwell, and Bastian, 2021; Vassilev et al., 2025).

2.10 Chapter 2 Summary and Identified Research Gap

This chapter presented a critical literature review related to continuous and traditional security assurance in smart-home IoT contexts. The main findings are:

- **Persistent limitations across traditional and CSA approaches:** Traditional methodology is plagued by fixed baselines, disjointed detection solutions, and reactive, event-driven paradigms (Pang et al., 2021; A. Shukla et al., 2022). Continuous methodologies reduce some of these issues but remain challenged by semantic in-

terpretation, drift-attack discrimination, and multi-layer correlation (Sommer and Paxson, 2010; Andresini et al., 2021; Aparcana-Tasayco, Deng, and Park, 2025).

- **Lack of unified, continuous, multi-layer security assurance:** Previous research mainly focuses on profiling, anomaly detection, or IDS separately, lacking an integrated framework that continuously evaluates the overall security posture (A. Shukla et al., 2022).

- **The Sys-Net blind spot:** A gap in combined system-network awareness prevents detection of intricate adversarial behaviours across layers, as most research focuses on either network-level or host-level monitoring (Lo et al., 2026; Bridges et al., 2020).

- **Black-box interpretability deficit:** The dominant use of opaque machine learning models in IoT anomaly detection and IDS prevents semantic threat attribution and adversarial robustness verification (Samek, Montavon, Vedaldi, et al., 2019; Neupane et al., 2022; Vassilev et al., 2025).

- **Motivation for the proposed framework:** The identified gaps justify a unified continuous security assurance framework that combines profiling, anomaly detection, and IDS with STRIDE-aligned threat interpretation to enhance detection fidelity, explainability, and zero-day resilience (Ardagna et al., 2017; Imran et al., 2026).

The findings of this literature review were directly used in the methodological design of this thesis. Hybrid Sys-Net profiling motivated by identified lack of cross-layer visibility, profiling-grounded anomaly assessment motivated by limitations on isolated anomaly detection, STRIDE-aligned classification motivated by weak semantic interpretation, and the limited consideration of adversarial evasion motivated the physical consistency verification and the continuous security-posture developed in the subsequent methodology.

Chapter 3

Methodology

3.1 Introduction

Smart-home IoT security sits at an awkward intersection: the devices that need protecting are numerous, heterogeneous, and resource-constrained, yet the threat landscape they face is as varied as it would be for any networked system. The literature reviewed in the preceding chapter reveals a structural gap in how the field has responded. Most deployed solutions monitor either the network layer or the host layer, but not both together, and those that do combine telemetry rarely do so in a way that allows the two streams to illuminate each other. The consequence is a class of attacks that remains systematically invisible, activity that is unremarkable from a network perspective but anomalous from a host perspective, or vice versa.

This chapter sets out the methodological response to that problem. The work was organised around three preparatory phases before any detection experiment could be run: constructing a physical testbed that realistically represents a heterogeneous smart-home; generating and collecting a dataset, SmartHome-StrideDS, that captures both normal operation and adversarial activity with STRIDE-aligned labels; and characterising the baseline vulnerability posture of each testbed device. These phases are described in the sections that follow with enough procedural detail to permit independent replication.

At the centre of the proposed work stands the Continuous Security Assurance (CSA) framework, a four-phase detection pipeline designed to move from raw telemetry to structured threat intelligence in real time. The remainder of this chapter explains how

that pipeline is constructed, how the experimental environment was configured to test it, and how the evaluation procedures were designed to yield practically meaningful results.

3.2 Problem Formulation and Solution Concept

3.2.1 The Sys-Net Blind Spot in Smart-Home Security

The dominant model in smart-home intrusion detection treats the network and the host as separate, self-contained objects of study. A Network Intrusion Detection System (NIDS) watches packets and flows; a Host-based Intrusion Detection System (HIDS) watches processes and file system state. Neither architecture formally models what happens when system-level behaviour and network-level behaviour are jointly examined, when the question is not "is this packet suspicious?" or "is this process suspicious?" but rather "is what the host is doing consistent with what the network is doing?"

This gap is not merely theoretical. ARP poisoning, for instance, produces clear network-layer anomalies, routing table distortions, unexpected MAC-to-IP associations, while leaving host resource metrics entirely undisturbed. A HIDS watching CPU, memory, and disk sees nothing unusual. Conversely, a living-off-the-land tampering attack that uses legitimate system binaries to modify configuration files produces host-level state changes without generating any unusual network traffic. A NIDS sees nothing unusual. Both attacks evade their respective single-plane detectors by exploiting the same structural property: the detector is blind to the telemetry plane in which the attack is expressed.

This thesis labels that structural property the **Sys-Net Blind Spot**. It arises from three interacting conditions. First, security solutions are deployed as either NIDS or HIDS, without any formal model of the relationship between the two planes. Second, security-relevant events in residential IoT environments routinely span both tiers simul-

taneously, a network event without a corresponding host explanation, or a host change without any network justification, is often more informative than either observation alone. Third, modern adversaries are well aware of single-plane monitoring and design their techniques accordingly, constraining their observable footprint within whichever plane the defender is watching.

The hybrid Sys-Net framework proposed here closes this gap by modelling both planes jointly. Cross-layer consistency analysis becomes possible, and the detection problem shifts from "does this individual measurement exceed a threshold?" to "does the combination of what the host and the network report make physical and operational sense?"

3.2.2 Architecture of the Continuous Security Assurance Framework

The proposed Continuous Security Assurance (CSA) framework is a series of four subsequent phases that take the result of a previous phase and convert it to a more semantically relevant expression of the detection and assurance problem. Hybrid behavioural profiling performs the first step of converting the raw data of the Sys-Net telemetry signals into per-device statistical baselines, creating a normal operating range for each of the features monitored. The second phase uses these baselines to flag observations as anomalies, and finds if an observation shows any meaningful difference from the established baselines. The third stage is anomaly-aware intrusion detection, in which anomalies are classified and then described (as a STRIDE element) as a likely attacker goal. The fourth one is Continuous Security Assurance which verifies the trustworthiness of the output of the classifier and hence the security evidence before accepting this. Each sample is submitted to three consistency checks, which are Temporal Statistical Consistency (T_{EWMA}), Physical Causal Consistency (C_{corr}), and STRIDE Semantic Viability (S_{STRIDE}). These checks ensure that the sample retains the expected statistical behaviour of the sampled

class, contains the same causal or correlating relationships among other features as the class, and has the desired operational properties of the predicted STRIDE attack. The same sample is subsequently fed through two more Isolation Forest gates and compared to the distribution of the source attack class, and to clean (normal) traffic. This offers a second level of assurance about the possibility of an adversarial sample being classified as "outside" of the two distributions, i.e. an "attack" sample, being misclassified by “outside”, i.e. a random forest, or “outside”, i.e. rule-based, checks. The three consistency checks and two Isolation Forest gates are fed into the master decision gate to detect and block a classifier-level evasion, which is a safeguard to prevent this type of evasion from making the security evidence less secure and from affecting the results of the three consistency checks.

Therefore, the four phases do not constitute a one-shot sequence, but rather form a security assurance pipeline: profiling establishes expected behaviour, anomaly detection identifies significant deviations, STRIDE-based intrusion detection categorises the detected anomalies, and the CSA verification phase verifies the physical and statistical consistency of the classification before it becomes part of the security posture. A high-level overview is presented in Figure 3.1.

3.3 Research Framework

3.3.1 Phase 1 — Hybrid Profile Construction

The profiling phase is responsible for producing a quantitative picture of how each device on the network behaves during normal operation. Two data streams are collected simultaneously and then combined to form the hybrid Sys-Net profile.

The host-level stream covers the following features: `cpu`, `memory`, `disk`, `load`, `uptime`, `tcp_conn`, and `open_ports_count`. These metrics give ground-truth visibility

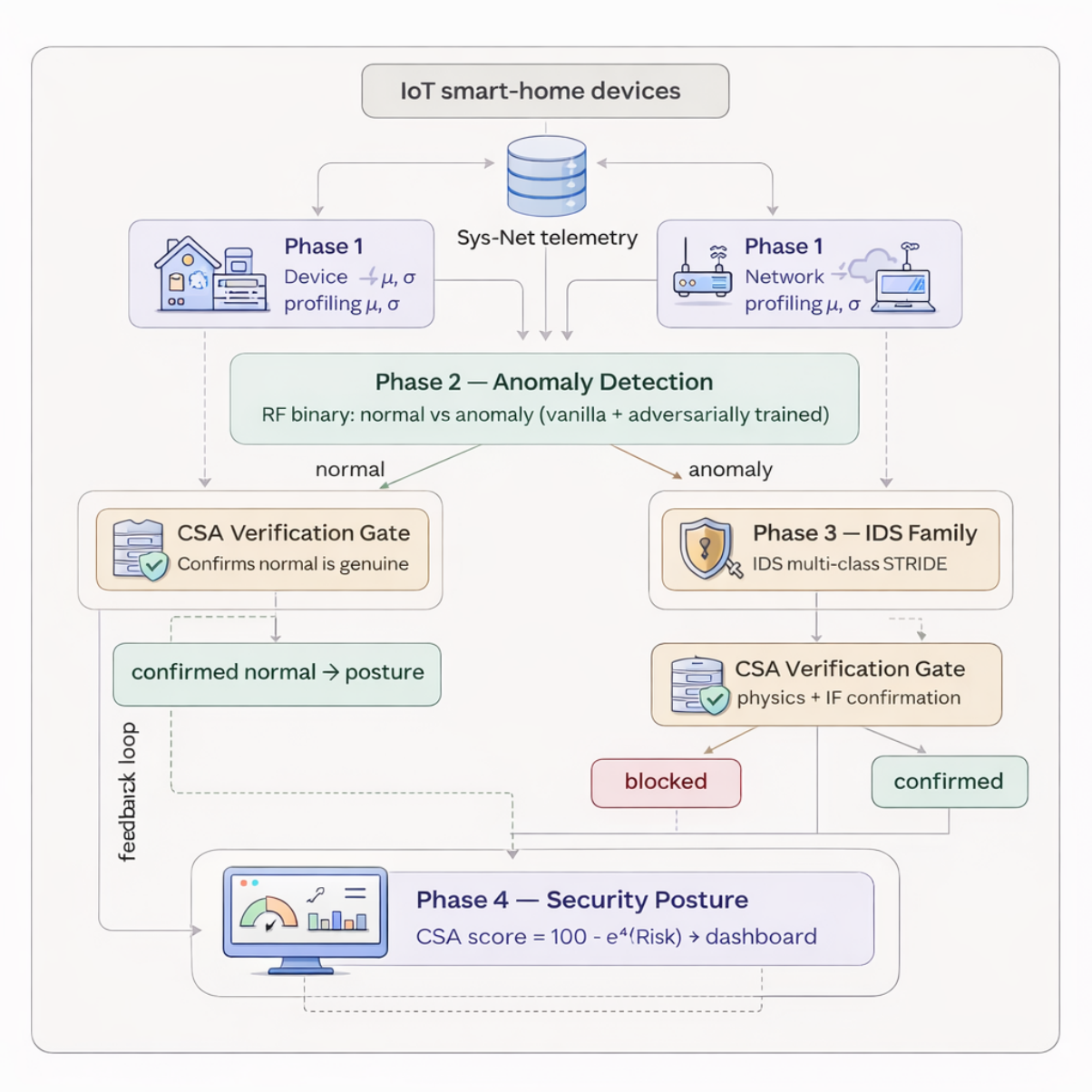

Figure 3.1: High-Level Architecture of the Continuous Security Assurance (CSA) Framework

into a device's internal computational state. Resource exhaustion attacks, malware execution, and other on-device compromise indicators that leave no trace at the network layer are observable here. The network-level stream covers `bytes_sent`, `bytes_recv`, `icmp_packets`, and `latency`. These features characterise the device's external communication behaviour and flag data exfiltration, reconnaissance, and denial-of-service conditions that may not disturb host metrics at all.

Raw telemetry from both streams is processed to expose temporal trends and inter-

feature dependencies rather than simply reporting instantaneous values. The aim is to capture the covariance structure of normal operation, the fact that, for a given device type, certain patterns of co-movement among features are characteristic of legitimate activity and deviations from those patterns are informative. The output of the profiling phase is a set of per-device, per-feature statistical models representing the expected operational range under benign conditions.

Adversarial Augmentation of Training Data

To make the downstream classifier resistant to bounded perturbations of its input, the training partition is augmented with adversarially perturbed instances before any model is fitted (Goodfellow, Shlens, and Szegedy, 2015; Biggio and Roli, 2018). For each of the five STRIDE-aligned classes, 10,000 synthetic samples are produced by adding per-feature uniform noise drawn from $[-\varepsilon_f, +\varepsilon_f]$, where the budget for feature f is:

$$\varepsilon_f = \alpha \times \big(\max(f) - \min(f)\big), \qquad \alpha = 0.15 \tag{3.1}$$

Every perturbed sample is clipped to the observed operational bounds of its feature so that the augmented data remains physically plausible (Vassilev et al., 2025). This procedure produces the adversarially trained classifier variant (`RF_advtrain`), which is evaluated in Chapter 5 alongside the clean-data variant (`RF_vanilla`). Comparing the two isolates the specific contribution of adversarial training to the overall defence, separate from the contribution of the CSA physical consistency verification layer.

3.3.2 Phase 2 — Profiling-Grounded Anomaly Detection

The anomaly detection layer takes the per-device profiles produced in Phase 1 and applies them to incoming real-time observations, issuing a binary judgement , normal or anomalous, for each data point. The detection engine is a Random Forest classifier,

selected because it handles the noise characteristics of IoT telemetry well, resists overfitting on high-dimensional Sys-Net feature vectors, and achieves 99.92 % test accuracy on SmartHome-StrideDS (Géron, 2022, see also Section 6.3).

The model produces a normalised deviation score that quantifies the distance between each incoming observation and the device-specific operational envelope. Observations whose score exceeds a calibrated threshold are flagged for further processing; those that fall within the envelope are discarded without any additional computation. This gate is an intentional design choice: restricting downstream classification to statistically confirmed anomalies keeps the pipeline efficient on resource-constrained hardware and reduces the volume of material that the more computationally expensive IDS layer must handle.

The sequential operation of the proposed profiling-grounded anomaly detection component is shown in Figure 3.2. First, system-level and network-level telemetry are synchronised and then compared with the device-specific statistical baseline using both the mean and standard deviation. The resulting deviation score indicates the deviation from expected behaviour, and the binary Random Forest classifier determines whether the observation is normal or anomalous. Any observations found to be anomalous are passed to the STRIDE-aligned intrusion detection component for further analysis. Normal observations pass through the CSA verification path, thus keeping the security posture on the monitoring dashboard up to date. This process illustrates how behavioural profiling can be used to aid in the identification of anomalies prior to threat classification.

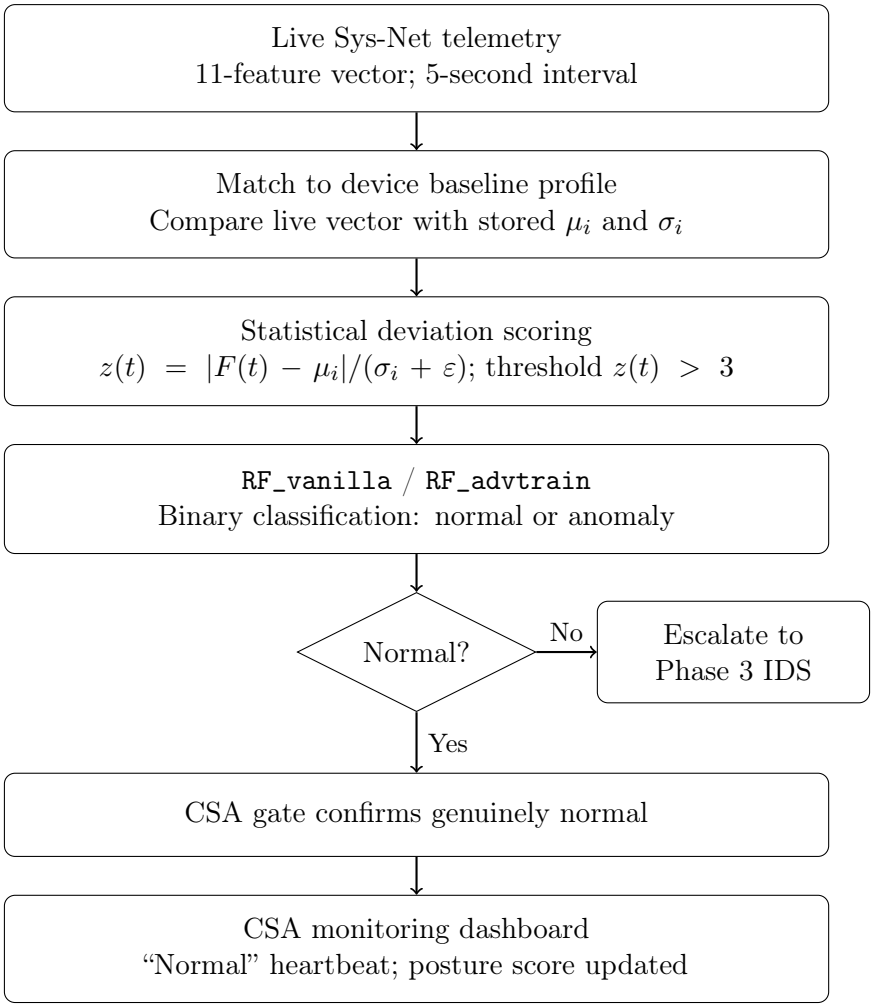

Figure 3.2: Profiling-Grounded Anomaly Detection Workflow

3.3.3 Phase 3 — Anomaly-Aware IDS Model

Where the anomaly detection layer asks whether something unusual has occurred, the IDS layer asks what kind of unusual thing it is. A multi-class Random Forest classifier maps each confirmed anomaly to a STRIDE threat category, using the full Sys-Net feature vector as evidence.

The available training data associate feature patterns with particular threat categories. A sustained, co-occurring elevation of `cpu` and `load` was associated with a Denial of Service-labelled observation (Gavric, Prasad Bhandari, and Shalaginov, 2026). A rapid increase in `open_ports_count` is associated with reconnaissance observations labelled as

Information Disclosure. Anomalies detected through `tamper_file_check` are associated with Tampering-labelled observations (Chen et al., 2023). Other mappings require joint evidence from multiple features: `bytes_sent` and `bytes_recv` are interpreted against the device-specific throughput baseline rather than any absolute threshold, and `tcp_conn`, `icmp_packets`, and `latency` are evaluated together to distinguish among spoofing, flooding, and information leakage, since no single one of these features is diagnostic on its own.

After classification, each labelled sample passes through the CSA physical consistency verification layer, which checks whether the feature values associated with the predicted threat class are physically coherent with the operational requirements of that attack type. The formal specification of this layer, including its governing equations and master detection gate, is presented in Chapter 4, Section 4.14.

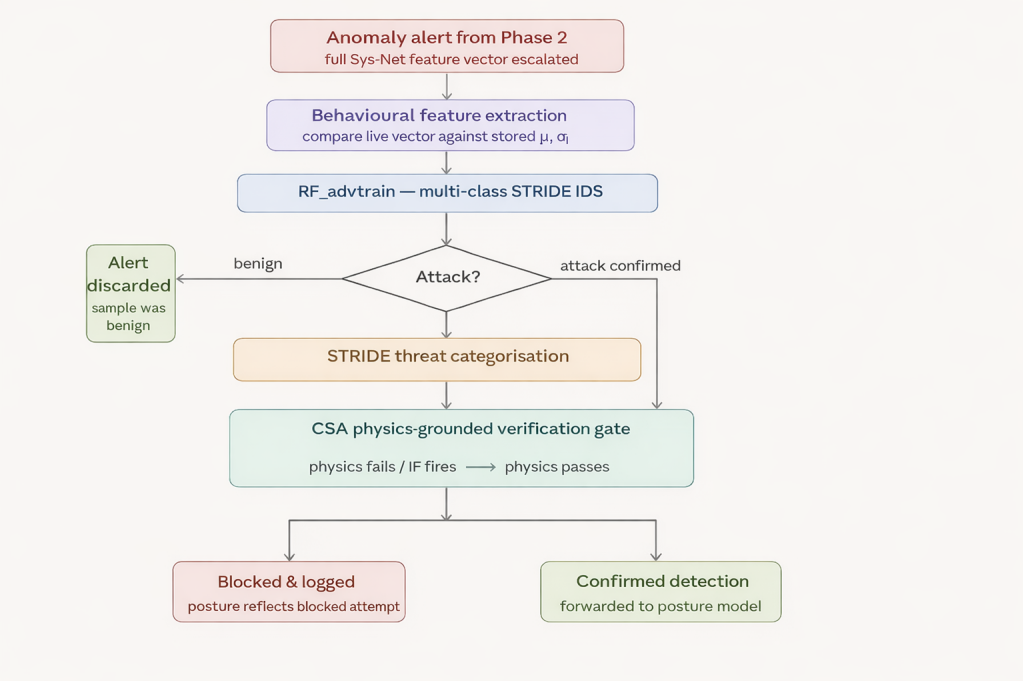

Figure 3.3: IDS Classification Flow Diagram

3.3.4 Phase 4 — Continuous Security Assurance

Phase 4 provides Continuous Security Assurance by verifying the trustworthiness of classifier output before it contributes to the security posture. As introduced in Section

3.2.2 and formalised in Section 4.14, the verification stage combines Temporal Statistical Consistency, Physical Causal Consistency, STRIDE Semantic Viability, the source-class Isolation Forest, and the normal-class Isolation Forest through the master decision gate. Continuous operation of this assurance process is supported by three background recalibration mechanisms: adaptive sensitivity adjustment, monitoring interval tuning, and baseline profile adaptation.

Adaptive sensitivity adjustment runs every thirty minutes. The system examines the false-positive rate from recent IDS alerts and the percentile shift in each feature's behavioural trend. Features whose distributions have been stable under benign conditions have their anomaly index $A_i(t)$ dampened by a factor $\alpha < 1$; features that are showing elevated variance or a rising anomaly rate receive a sensitivity boost $\beta > 1$. Both adjustments are anchored to the per-device behavioural envelope established during profiling, using the operational $k = 3$ significance boundary.

Monitoring interval tuning runs every five minutes. A device stability score is computed from the variance of recent Sys-Net observations and the rate of recent IDS alerts for that device. Devices flagged as high-criticality are queried at ten-second intervals; devices that have been consistently stable revert to sixty-second intervals. This dynamic allocation of sampling budget concentrates measurement effort where it is most needed.

Baseline profile adaptation updates the per-feature parameters incrementally using a sliding-window exponential moving average, allowing the profiles to track genuine long-term shifts in device behaviour without being contaminated by sustained attack traffic:

$$\mu_i^{\text{new}} = (1 - \lambda)\, \mu_i^{\text{old}} + \lambda\, \bar{f}_i(t) \tag{3.2}$$

$$\sigma_i^{\text{new}} = \sqrt{(1 - \lambda)\big(\sigma_i^{\text{old}}\big)^2 + \lambda\big(\bar{f}_i(t) - \mu_i^{\text{new}}\big)^2} \tag{3.3}$$

where $\bar{f}_i(t)$ is the windowed mean of feature f_i over the most recent observation

window and $\lambda \in [0, 1]$ controls how quickly the baseline adapts to new data (Gama et al., 2014).

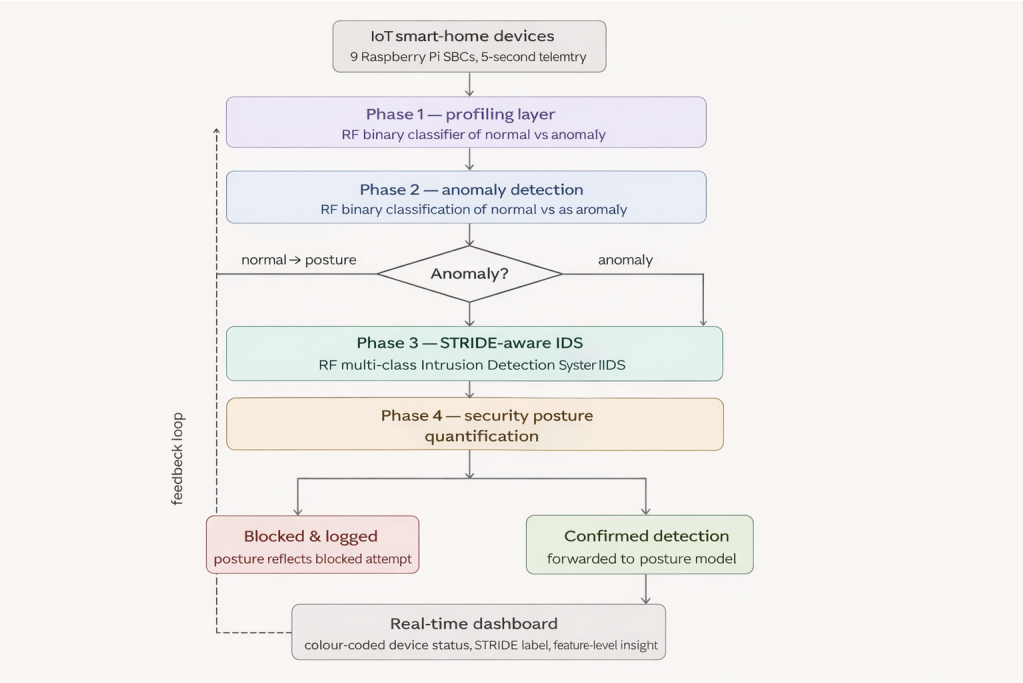

Figure 3.4: CSA Monitoring Loop

3.4 Experimental Setup

3.4.1 Hardware and Software Configuration

Nine Raspberry Pi single-board computers form the backbone of the testbed. This platform was chosen for the same reasons it appears throughout IoT security research: it runs a full Linux environment, it is configurable enough to emulate a wide range of device roles, and its computational profile is representative of the resource-constrained nodes found in residential deployments (Jawad et al., 2026; Jeyasoundari et al., 2026). To reflect the operating-system heterogeneity typical of real smart-home installations (Vardakis et al., 2024), five of the nine devices run Raspbian 5 and the remaining four run Ubuntu. Peripheral hardware was deployed to simulate the sensor and actuator devices found in a working home: three ArduCam modules and two Pi Camera units for video surveil-

lance, one robotic arm, and three thermostat sensors providing continuous temperature readings.

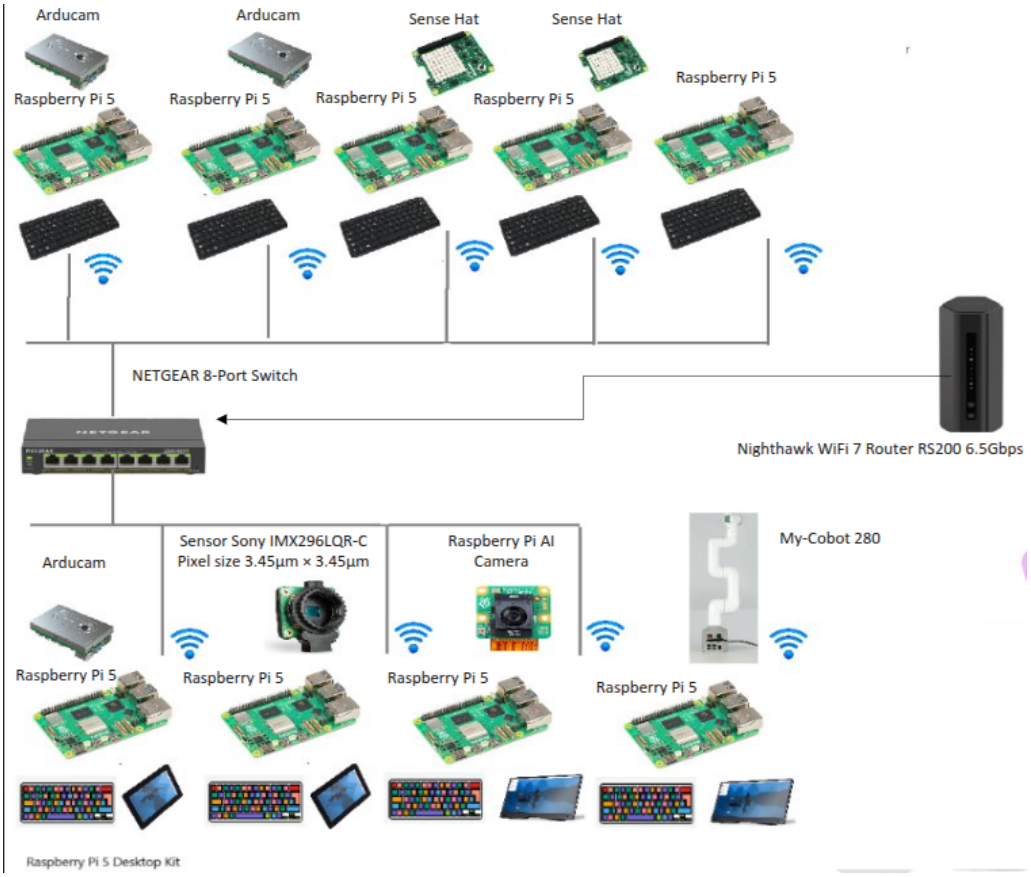

Figure 3.5: The testing network containing the Raspberry Pi devices and associated peripherals

3.4.2 Network Architecture

The network topology mirrors a standard residential configuration. A Netgear router handles general management and external internet access. Three of the nine SBCs are connected via Ethernet through a Netgear switch, providing a wired segment representative of fixed smart-home hubs; the remaining six SBCs and all peripheral devices connect over the wireless segment. All attack simulations were conducted within a physically and logically isolated environment, ensuring that experimental conditions were fully controlled and that no traffic could reach or affect systems outside the testbed.

3.4.3 CSA Consistency Verification Layer: Operational Overview

The consistency verification layer sits orthogonal to the primary Random Forest classifier. Rather than learning decision boundaries from data, it encodes physical and causal domain knowledge about IoT device behaviour, constraints that any realistic attack payload must satisfy regardless of how its feature-space representation has been manipulated. The layer therefore provides a second line of defence that is structurally immune to the statistical evasion strategies that threaten data-driven classifiers.

The layer comprises three complementary rules. The temporal consistency rule (T_{EWMA}) flags feature vectors whose EWMA-normalised z-scores deviate systematically from the class-specific operational baseline. The causal correlation rule (C_{corr}) enforces the directional co-movement relationships that are known to hold between causally linked feature pairs under normal and attack conditions alike. The STRIDE semantic consistency rule (S_{STRIDE}) verifies that each attack-class sample retains at least one characteristic signature feature above a minimum operational threshold, without which the attack would not, in practice, be capable of achieving its stated objective.

All three rules are evaluated over a fixed seven-dimensional feature subset: `cpu`, `memory`, `disk`, `load`, `tcp_conn`, `icmp_packets`, and `latency`. Two per-class Isolation Forest models complement the rule-based layer: a source-class forest (IF_{src}) trained on the unperturbed class distribution, and a normal-class forest (IF_{norm}) trained exclusively on clean benign traffic. A perturbed sample that evades the classifier may nonetheless fail the Isolation Forest check if it does not occupy the correct region of the joint feature space. The full formal specification, including the master detection gate, is presented in Chapter 4, Section 4.14.

Attack Model: Decision-Based Black-Box Attack

The adversarial evaluation adopts a decision-based black-box threat model (Papernot et al., 2017; Vassilev et al., 2025). The adversary receives only the predicted class label from the classifier and is denied any access to model weights, gradients, internal architecture, or the parameters of the CSA defence layer. This is a realistic and conservative threat model for residential deployments, where an attacker probing a deployed system would have no more access than the label that the system's response implies.

Within this threat model, the oblivious variant tries to shift the classifier's output without knowing that a physical consistency layer exists. The adaptive variant knows about all three consistency rules and penalises constraint violations in its loss function, representing the strongest attacker that can be mounted within this threat model. Comparing the two directly measures how much additional protection the consistency layer provides against an informed adversary.

3.5 Methodological Design Rationale and Novelty

3.5.1 Architectural Requirements and Design Alternatives

The four requirements of the proposed architecture are: achieving coherence between host and network behaviours; distinguishing ordinary behaviours from those of security interest; mapping device behaviours to security-relevant categories; and translating validated host and network behaviours into a usable and continuously updated security state. These requirements cannot be met independently by a standalone host-based monitoring system, a standalone network-based intrusion detection system, or an isolated machine-learning classifier (Ardagna et al., 2017; A. Shukla et al., 2022; Khraisat and Alazab, 2021). The challenges identified in Section 2.9 connect to the framework components: hybrid profiling addresses cross-layer visibility; profiling-grounded anomaly

detection addresses behavioural discrimination; STRIDE-aligned classification supports semantic interpretation; and the assurance and verification stages address continuous posture assessment and adversarial reliability.

The layered architecture is designed to highlight the trade-offs identified in related work. While device-oriented mechanisms provide complementary information about device behaviour, they may lack the necessary network-level context for detecting network-mediated attacks. On the other hand, network-oriented approaches can provide valuable traffic information but cannot directly detect host-level execution or resource modifications without additional telemetry information. This introduces a requirement for synchronising data obtained from the network and the devices under analysis, enabling their combined use for anomaly escalation, device-specific calibration, and classification of security-relevant observations across multiple processing stages (Ben Slimane, Abd-Elkawy, and Maqbool, 2024; Meidan et al., 2023).

3.5.2 STRIDE Alignment and Threat Interpretation

STRIDE is an acronym representing a taxonomy of threats that identifies and classifies security threats according to their effects on a system. It is not an anomaly-detection algorithm and does not independently assess the nature of an observed telemetry vector. The profiling-based detection module is responsible for anomaly detection, while the trained intrusion-detection module, which classifies intrusions using labelled system-level and network-level observations, assigns supported attack categories. STRIDE provides the semantic framework for interpreting and communicating those classification outcomes (Shevchenko et al., 2018).

As a result, the relationship between observable telemetry and STRIDE threat categories cannot be assumed to be uniquely causal. For example, a sustained rise in processor usage from 90% to 100% may be related to resource exhaustion but may also be caused by a legitimate, continuously running high-throughput application. Similarly,

abnormal traffic volume may be related to Denial of Service, reconnaissance, or an authorised high-volume application. The association of such indicators with a particular threat category is therefore treated as evidence of an attack scenario within a given training dataset, rather than as a strict indicator unique to that threat category.

The rationale for choosing STRIDE was that the principal use cases at this stage of the proposed architecture involve understanding the category of an identified threat rather than assigning a CVSS or DREAD risk-severity value. DREAD and CVSS are not inherently unsuitable security frameworks; rather, they serve different analytical purposes from the threat-category interpretation provided by STRIDE (Shevchenko et al., 2018; Shostack, 2014).

Table 3.1: Comparison of Threat-Modelling Approaches

Model	**Primary Purpose**	**Role and Boundary**
STRIDE	Threat categorisation	Categorises supported threat types after detection; it is not an anomaly detector (Shevchenko et al., 2018)
DREAD	Qualitative risk assessment	Supports risk prioritisation rather than runtime telemetry classification (Shostack, 2014)
CVSS	Vulnerability severity scoring	Scores vulnerability severity; it does not classify observed behaviour
Attack Trees	Design-time attack-path analysis	Supports architectural threat analysis, not runtime IDS classification (Shostack, 2014)
MITRE ATT&CK	Adversary tactics and techniques knowledge base	Supports behaviour-to-technique analysis; it is not a standalone anomaly detector (MITRE, 2026)

3.5.3 Intrusion Detection Model Selection

Random Forest was selected for the intrusion-classification component because of its ability to model non-linear interactions between input features and handle a heterogeneous

feature representation. These characteristics are relevant to the available hybrid feature representation and the requirements of the CSA framework (Breiman, 2001). The suitability of Random Forest is further examined in Chapter 5 through a controlled comparison with the other selected classification approaches using the proposed dataset. The results of this comparison are used to assess its effectiveness in classifying the observed behavioural and traffic patterns (Table 5.3).

Linear classifiers provide comparatively simple decision functions but may not demonstrate the strongest detection performance in scenarios involving non-linear relationships between features. Neural-network and deep-learning approaches enable the learning of richer representations, but they are not necessarily more accurate, explainable, or adversarially robust than all alternatives. In particular, feature-importance measures do not imply causal explanations, and the adversarial performance of the physical consistency verification layer must be assessed separately from classifier performance (Breiman, 2001; Neupane et al., 2022; Samek, Montavon, Vedaldi, et al., 2019).

3.5.4 Architectural Novelty and Methodological Boundaries

The novelty of the proposed framework does not lie in the Random Forest algorithm or the STRIDE taxonomy, but in the way established mechanisms are arranged and assessed as a continuous security assurance architecture. The anomaly-detection step determines whether observations trigger escalation; the intrusion-detection step allows the assignment of supported threat categories; and the consistency-verification step checks whether the available evidence is consistent with temporal, causal, and semantic constraints. The novelty claim is limited to this thesis's specific implementation and evaluation of those mechanisms and the way available security evidence is translated into a continuously updated posture representation. This is not a claim that the general integration of host/network monitoring, STRIDE classification, and posture management is unprecedented: Imran et al. (2026) also propose these functions and report a case study

(Imran et al., 2026).

The empirical threat coverage is limited to four of the six STRIDE categories. The profiling and classification models are primarily evaluated using a Linux-based Raspberry Pi smart-home testbed, while the adversarial evaluation is limited to the threat models and feature-space perturbations documented in the experiments. The use of physical consistency constraints also requires domain-specific calibration when transferring the framework to external datasets, which precludes universal claims of attack detection or adversarial robustness.

SmartHome-StrideDS supports this design by synchronising host- and network-level observations and mapping the evaluated attack labels to STRIDE categories. Its evaluation remains limited to the four implemented attack categories and the documented testbed and experimental procedures (Ring et al., 2019).

3.6 Evaluation Design and Procedures

To distinguish between these research claims, the evaluation uses controlled within-dataset comparisons, followed by restricted cross-dataset anomaly-detection experiments, and conducts adversarial assessments under the two specified attacker models: oblivious and adaptive. The datasets, experimental partitions, validation procedures, evaluation metrics, and comparison criteria used to produce the results reported in Chapter 5 are described in the following subsections.

3.6.1 Datasets, Feature Preparation, and Experimental Partitions

The primary dataset is SmartHome-StrideDS, which combines synchronised host-level process metrics and network-level traffic records. TON-IoT Linux process data provide a

restricted process-level condition for anomaly detection; the TON-IoT host-process and network-flow corpora are used in the documented adversarial evaluations. For binary anomaly detection, labels are Normal (0) and Attack (1). Where a train-test partition is used, the data are divided 80/20 with stratified sampling to preserve class proportions, and features are standardised using training-set statistics before model fitting.

3.6.2 Supervised Classification and Anomaly-Detection Protocols

For supervised STRIDE classification, the SmartHome-StrideDS partition retains the Normal class and the four evaluated attack categories: Denial of Service, Spoofing, Tampering, and Information Disclosure. The held-out evaluation uses the documented stratified 80/20 partition. At runtime, observations flagged by the profiling-based anomaly-detection stage are forwarded to the intrusion classifier, as described in Chapter 4. The multiclass classifier evaluation is distinct from binary anomaly detection; the TON Linux Process comparison does not represent transfer of the SmartHome-StrideDS multiclass classifier (Géron, 2022).

Vulnerability assessment and sensitivity thresholds. Every device in the testbed underwent an automated vulnerability scan before experimental data collection. The scan covered open network ports, exposed service banners, default credentials, and out-of-date firmware versions. Each identified weakness was scored using the Common Vulnerability Scoring System (CVSS), which assigns a severity between 0 and 10 on the basis of exploitability, impact, and access vector.

This static vulnerability characterisation is integrated into the dynamic posture model through the STRIDE sensitivity threshold ϕ_j. Devices that carry a higher aggregate CVE score, meaning they have more known and unpatched weaknesses, are assigned a lower value of ϕ_j, which tightens the anomaly- detection sensitivity applied to their

traffic. The reasoning behind this asymmetry is that a poorly hardened device is at once more likely to be targeted by a capable adversary and more consequential as a point of entry if a detection is missed. Coupling static vulnerability metadata to dynamic detection thresholds in this way introduces risk-proportionality compared with schemes that apply the same sensitivity to every device regardless of its vulnerability surface.

3.6.3 Evaluation Metrics and Comparison Criteria

For binary anomaly detection, the reported metrics are Accuracy, Precision, Recall, F1-score, and False Positive Rate, which capture sensitivity and false-alarm behaviour. For multiclass intrusion detection, per-class and macro-averaged F1-scores are reported across the four implemented STRIDE categories. F1-score is also used as the comparison criterion for the selected classifier baselines.

3.6.4 Modality Ablation and Baseline Classifier Comparisons

Four classifiers were selected for comparison on SmartHome-StrideDS: Logistic Regression, Random Forest, Support Vector Machine (SVM with an RBF kernel), and Multilayer Perceptron (MLP). Their held-out comparison uses the same stratified 80/20 partition (Table 5.3). The modality ablation separately compares hybrid, network-only, and host-only Random Forest configurations using the same labelled rows, split, and classifier hyperparameters; the reported SmartHome-StrideDS comparison uses 500 estimators and the Gini criterion (Table 5.5). The TON-IoT single-plane comparisons are treated separately and are not interpreted as equivalent modality-ablation conditions.

3.6.5 Adversarial Threat Models and Perturbation Procedures

Adversarial perturbations are applied to correctly classified samples from the 20% held-out partition. The *oblivious attacker* applies bounded perturbations to evade the classifier

without knowledge of the CSA defence layer. The *adaptive attacker* incorporates explicit penalties for violations of the three CSA consistency rules, T_{EWMA}, C_{corr}, and S_{STRIDE}, weighted at 2.0, 3.0, and 2.0, respectively, into the attack loss function (Papernot et al., 2017; Vassilev et al., 2025).

In both cases, the per-feature perturbation budget is

$$\varepsilon_f = \alpha \times \big(\max(f) - \min(f)\big), \tag{3.4}$$

with $\alpha \in \{0.15,\ 0.10,\ 0.05,\ 0.01\}$. The attacks are decision-based and black-box: the attacker has no access to model weights, gradients, or training data. `RF_vanilla` is trained on clean data; `RF_advtrain` is trained on `train_augmented.csv`, which supplements the original training corpus with 10 000 adversarially perturbed instances per class at $\alpha = 0.15$.

3.6.6 Cross-Dataset Evaluation and Validity Boundaries

TON Linux Process data are used as a restricted process-level comparison for binary anomaly detection under limited telemetry availability. This comparison does not constitute transfer of the SmartHome-StrideDS multiclass classifier. The documented cross-dataset results do not establish generalisation to all IoT devices or environments. They are reported in Chapter 5, Section 5.9.

The adversarial experiments are additionally conducted on the TON-IoT network and host-process corpora to examine the portability of the verification approach under different feature representations. Unless otherwise stated for a particular experiment, the reported classifier metrics are point estimates derived from the documented held-out test partitions; they should not be interpreted as repeated-run averages or accompanied by measures of statistical uncertainty that were not calculated.

3.7 Chapter Summary

This chapter has laid out both the conceptual foundation and the empirical realisation of the proposed CSA framework. The central argument is that single-plane IoT security monitoring leaves a structural blind spot that adversaries can and do exploit, and that closing it requires the joint modelling of host-level and network-level telemetry. The CSA framework addresses this through a four-phase pipeline:

$$\text{Profiling } (\mu_i, \sigma_i) \;\rightarrow\; \text{Anomaly Detection } (A_i, A) \;\rightarrow\; \text{STRIDE IDS Mapping } (S_j)$$
$$\rightarrow\; \text{Continuous Recalibration}$$

The profiling phase establishes device-specific statistical baselines from benign telemetry. The anomaly detection phase applies those baselines in real time, scoring deviations and filtering the data stream before it reaches the classifier. The IDS mapping phase converts confirmed anomalies into structured threat intelligence aligned with the STRIDE model. The continuous assurance phase keeps the first three phases calibrated as operational conditions evolve.

To test this framework, a nine-device smart-home testbed was constructed and instrumented, and the SmartHome-StrideDS dataset was collected under both benign and adversarial conditions. The evaluation design, including stratified hold-out testing, a restricted cross-dataset anomaly-detection comparison on an independent benchmark, and oblivious and adaptive adversarial robustness assessment, was chosen to assess performance under the documented conditions. The material presented here collectively addresses RQ1 through RQ4 and provides the experimental and theoretical foundation on which all subsequent chapters rest.

Chapter 4

Design and Implementation of the Continuous Security Assurance Framework

Introduction

In this chapter, the design and implementation of the Continuous Security Assurance framework to address the research challenges identified in Chapter 2 and the architectural requirements established in Chapter 3 are presented. It starts with the SmartHome-StrideDS dataset and the creation of synchronised host-level and network-level feature representations and progresses to the design of behavioural profiles for each device and the development of profiling-based anomaly detection. The chapter then describes the STRIDE-based intrusion classification mechanism, the physical consistency verification layer, and the security posture quantification model. Lastly, it explains the operation of the real-time dashboard and the integrated processing pipeline. Throughout, the focus is on the implementation and interaction of the framework components, while their performance, comparative evaluation, and robustness are presented in the experimental evaluation in Chapter 5.

4.1 Hybrid Behavioural Profiling

Anomaly detection is, at bottom, a comparative exercise: it can only proceed once someone has decided what counts as normal. In a heterogeneous residential network, that decision is far from straightforward. A Raspberry Pi acting as a surveillance hub produces a telemetric signature that is essentially unrelated to that of an identically specified node serving as a thermostat gateway; the workload rhythms, connection patterns, and resource demands of the two roles diverge in almost every dimension. Applying a shared population-level detection boundary to both nodes simultaneously would produce thresholds that are too conservative for one and too permissive for the other. The practical implication is unambiguous: wherever device roles differ, detection boundaries must be derived separately, from the observed behaviour of each individual node under confirmed benign conditions.

The profiling component fulfils the first research objective by constructing such device-specific baselines for the nine-node smart-home testbed described in Chapter 3. The work proceeds in four stages. The SmartHome-StrideDS dataset is assembled by collecting synchronised host-level and network-level measurements across both normal and adversarial operating conditions. A principled feature selection procedure identifies the eleven metrics that jointly satisfy the study's observability, deployability, and temporal-alignment requirements. A statistical profiling model is then fitted separately to the benign observations of each device, yielding per-device mean and standard-deviation estimates that define the probabilistic envelope within which normal operation is expected to fall. Finally, a validation experiment comparing this per-device design against a conventional population-level approach quantifies the practical gain attributable to device-specific profiling. The resulting profiles serve as the calibration baseline for the anomaly detection layer of Section 4.7 and the intrusion detection layer of Section 4.10.

4.2 The SmartHome-StrideDS Dataset

SmartHome-StrideDS is the principal dataset contribution of this thesis. Its defining characteristic is the temporal alignment of host-level process telemetry with network-level traffic measurements at every sampling instant, across a nine-device testbed operating under both unattacked and adversarial conditions. This design was motivated by a specific gap in the existing literature: no publicly available IoT security corpus at the time of collection combined both telemetry planes under STRIDE-aligned ground-truth labels (Ring et al., 2019).

4.2.1 Data Collection

Collecting measurements from nine physically distributed single-board computers without deploying persistent monitoring agents on any of them required a bespoke collection architecture. Persistent agents were ruled out on practical grounds: on resource-constrained IoT nodes, a resident monitoring process would itself perturb the CPU and memory readings it was there to observe, introducing measurement artefacts that would contaminate the very baselines the study depended on. Instead, a Python orchestrator was written that opens SSH sessions to each device through the Paramiko library, dispatches a tightly scoped set of measurement commands, harvests the output, and closes the connection before the next polling cycle begins. Nine threads run these sessions concurrently, completing the full round-trip within a five-second window for all nodes simultaneously. That five-second granularity was not arbitrary: it was short enough to capture the rapid onset of denial-of-service conditions and ARP-level routing distortions, while long enough to avoid a situation where collection traffic itself constituted a measurable fraction of each node's network load.

Two families of metric are recorded at every cycle. The first covers the internal computational state of the device. CPU utilisation, memory consumption, disk occu-

pancy, and the one-minute system load average collectively characterise how hard the processor and storage subsystems are working at each sampling instant. Uptime provides a complementary stability indicator, since abrupt reductions in this counter are a reliable signature of attacker-induced crashes or persistence mechanisms that require a controlled reboot to activate. The count of active TCP connections and open listening ports rounds out the host group, capturing both session activity and service exposure. The second family describes how the device interacts with the network. Bytes transmitted and received on the primary interface track throughput behaviour; the ratio and absolute magnitude of these two counters distinguish exfiltration-style activity from symmetric flooding. ICMP packet volume registers ping-based scanning and network-layer flood attempts. Round-trip latency to a fixed reference host serves as an indirect measure of both network congestion and host-level processing delays that propagate outward into observable response times.

Every observation was written to a shared CSV file with a device identifier and a wall-clock timestamp attached at the moment of collection, producing a dataset in which each row corresponds to a specific device at a specific instant and carries measurements from both telemetric families in a single aligned record.

4.2.2 Attack Simulation and Data Labelling

Meaningful ground-truth labels require attacks that reflect the adversarial techniques actually observed in residential IoT deployments, not synthetic perturbations engineered to satisfy statistical convenience. The collection script therefore drove a structured sequence of scenario executions, with each phase producing observations tagged with the active attack class before being written to the corpus. Scenario design was anchored to the STRIDE threat taxonomy (Shevchenko et al., 2018), which offers a principled decomposition of adversarial intent into categories whose operational signatures are sufficiently distinct to support multi-class classification.

Normal Operation. The nine devices were first run under sustained unattacked conditions long enough to accumulate stable baselines. During this phase, the co-movement patterns between host and network metrics settled into the coherent rhythms that legitimate workloads produce: CPU loading tracked outbound byte rates in predictable ways, latency varied within narrow bounds, and connection counts reflected the background session activity typical of each device role (Sivanathan et al., 2019; Gavric, Prasad Bhandari, and Shalaginov, 2026). These cross-layer regularities are what the profiling layer encodes as normal. The detection problem, viewed from this perspective, is not fundamentally about identifying large values in any single metric; it is about recognising when the expected dependencies between metrics have been disrupted.

Denial of Service. Three DoS variants were executed to exercise different exhaustion mechanisms. A TCP SYN flood, generated via `hping3`, saturated the connection table by leaving half-open connections unresolved. A UDP flood, scripted in Python using Scapy, delivered high-rate datagrams to randomly selected high ports. A slow-rate HTTP flood, implemented through a modified Slowloris tool, consumed server concurrency by holding connections open indefinitely without sending complete requests, a technique specifically designed to exhaust resources while generating traffic volumes low enough to evade rate-based detection. Each variant ran for five to ten minutes (Cambiaso et al., 2013).

Spoofing. ARP cache poisoning was used to redirect traffic between two nodes on the local segment. This scenario is of particular relevance to the hybrid profiling design because ARP manipulation is entirely a network-layer event: it modifies routing tables and MAC-to-IP bindings without touching the host's CPU, memory, or filesystem state (Abad and Bonilla, 2007). A host-only detector watching resource utilisation and process activity would observe nothing unusual. The attack is visible only because the network-layer measurements diverge from their expected relationship with host-layer measurements that remain entirely undisturbed.

Tampering. Living-off-the-land (LotL) techniques were used to alter device state

through standard system utilities, deliberately avoiding any purpose-built exploit or known-malicious binary (Lo et al., 2026). The significance of this choice for the detection task is that signature-based methods are structurally incapable of flagging it: no binary hash, no known-bad process name, no signature rule matches anything executed. What the host layer does record is the downstream effect on filesystem integrity and device configuration, and the absence of any corresponding network event at the same sampling instant provides the cross-layer inconsistency that makes the intrusion detectable under the hybrid design.

Information Disclosure. Rather than capturing live exfiltration, this scenario modelled the preparatory stage that precedes it: the opening of unauthorised listening services and additional network ports through which data will subsequently be extracted (MITRE, 2026). Port exposure changes are immediately observable in the host-level service inventory. Because the profiling model associates each device's legitimate port configuration with specific network communication patterns, the appearance of new services without any corresponding change in those patterns constitutes a cross-layer anomaly that the framework flags before a single byte of sensitive data has left the device.

Collection ran in automated seven-hour sessions and produced approximately 187,000 labelled observations spanning five classes: Normal, DoS, Spoofing, Tampering, and Information Disclosure. Table 4.1 maps each scenario to the corresponding MITRE ATT&CK technique identifier.

Table 4.1: Attack Scenarios and MITRE ATT&CK Mappings

Scenario	**MITRE ATT&CK ID**	**Technique Name**
DoS (Resource Saturation)	T1499 / T1498	Endpoint / Network Denial of Service
Spoofing (ARP MITM)	T1557.002	ARP Cache Poisoning
Tampering (LotL Host Manipulation)	T1059	Command & Scripting Interpreter
Info. Disclosure (Pre-Exfiltration)	T1133 / T1046	External Remote Services / Network Service Discovery

Generalisability of the DoS Simulation

The DoS scenario implemented here uses ICMP-based resource saturation, consistent with T1498 (Network Denial of Service) in the MITRE ATT&CK framework. What the classifier ultimately learns is the resource exhaustion pattern, the sustained, co-occurring elevation of CPU utilisation and system load alongside network metric anomalies, rather than any protocol-specific signature. Prior work indicates that Random Forest classifiers trained on resource-based features at five-second aggregation granularities generalise across DoS protocol variants, including ICMP, SYN, and UDP floods. Extending the testbed to include explicit SYN-flood and UDP-flood scenarios via `hping3` and `iperf3` is identified as a priority direction for future work.

4.2.3 Feature Selection Methodology and Exclusion Rationale

Feature selection was governed by four requirements that had to be satisfied simultaneously. A candidate metric was included only if it offered a direct, individually interpretable signal for at least one of the four evaluated STRIDE categories, a property termed STRIDE Observability. Dual-Plane Coverage required that at least one retained feature per threat category come from each of the two telemetric planes, ensuring that neither the host nor the network dimension was treated as dispensable. Deployability excluded any metric whose collection would require kernel module insertion, eBPF instrumentation, or deep packet inspection , all of which exceed the administrative and computational privileges realistically available on residential IoT hardware. Temporal Synchronisability required that each selected feature be collectible atomically within a single five-second SSH polling cycle, with no inter-sample timing drift that would misalign the host and network columns of the dataset.

Several candidate categories were assessed and rejected on these grounds. System call trace data collected via `strace` or the Linux audit framework imposes collection overhead that violates the Deployability criterion on the target hardware. ARP table state

is event-driven rather than polled, so it cannot be reliably captured within a fixed five-second window without introducing alignment uncertainty. Network packet inter-arrival time statistics require unbounded packet buffering that exceeds the available memory on the test nodes. DNS query logs are subject to resolver caching that introduces temporal misalignment between the time a lookup was issued and the time it appears in the log. Memory page fault rate was collectible but proved statistically indistinguishable across all five classes at five-second aggregation granularity, failing the Observability criterion. The eleven features that clear all four requirements represent the widest practically achievable STRIDE-observable and temporally aligned feature set for this deployment context.

4.3 Feature Engineering

Across the full 187,000-observation corpus, Normal-class records account for 53.5 % of the dataset and Attack-class records for the remaining 46.5 %. This near-parity was a deliberate design target rather than an incidental outcome of the collection schedule. When one class vastly outnumbers another, a naive classifier can achieve deceptively high accuracy by predicting the majority class unconditionally; minority-class recall collapses, and the metric that matters most for a security application, the rate at which genuine attacks are caught, deteriorates without any corresponding decline in the headline accuracy figure (He and Garcia, 2009). The collection protocol was therefore managed to prevent that imbalance from arising.

4.3.1 System-Level Features

Among the host-level features, the resource utilisation quartet, `cpu`, `memory`, `disk`, and `load`, tracks what the processing and storage subsystems are actually doing at each sampling instant. Their joint behaviour is more informative than any individual member: a CPU spike accompanied by a proportionate rise in load is characteristic of a flooding

attack or a cryptomining implant, whereas a CPU spike occurring without any corresponding change in outbound byte traffic more plausibly reflects local computation than a network-facing intrusion. Separating these scenarios requires each metric to be read against its context, which is precisely what cross-layer profiling enables. The `uptime` counter serves a distinct purpose: it is not a performance metric but a stability indicator. A device whose uptime counter resets unexpectedly has either crashed or been rebooted, and in an environment where no authorised maintenance was scheduled, either event warrants investigation, as a possible sign of resource exhaustion caused by a DoS attack, or of an adversary cycling the device to activate a persistence mechanism implanted during a prior compromise.

4.3.2 Network-Level Features

On the network side, `bytes_sent` and `bytes_recv` describe the throughput profile of the device at each sampling window. A pronounced asymmetry in which outbound bytes substantially exceed the device's established normal outbound-to-inbound ratio is a well-recognised indicator of data exfiltration; symmetric elevation of both counters is more consistent with bidirectional flooding. The pair `tcp_conn` and `open_ports_count` captures the device's connection footprint and service exposure. Rapid growth in open connection count, particularly when accompanied by ports absent from the device's baseline service inventory, is consistent with scanning activity, lateral movement attempts, or the service-staging behaviour that precedes an information disclosure event. `icmp_packets` registers ICMP-based activity; volumes that exceed the device's normal ping traffic indicate either a network sweep or participation in an ICMP-based flood. Finally, `latency`, measured as round-trip time to a stable reference host, captures network congestion from flooding attacks and also reflects host-side processing delays that propagate into observable response times when the device is under computational pressure.

4.3.3 Security Relevance of the Joint Feature Set

Taken individually, none of the eleven features is unambiguously diagnostic. A momentary CPU spike is as consistent with a scheduled backup job as with a denial-of-service attack; a brief rise in outbound bytes is as consistent with a firmware update as with exfiltration. The security content of the feature set emerges from the relationships among measurements rather than from the absolute magnitude of any one of them. A CPU elevation that co-occurs with anomalous TCP connection growth, unexplained new open ports, and above-normal latency presents an evidential combination of a qualitatively different character from any single deviation considered in isolation; the aggregate is far more discriminating than its parts. Beyond that, the most forensically significant signal in the dataset is often not an unusually large value but a cross-layer inconsistency: a host-level state change that arrives with no accompanying network explanation, or a network-layer anomaly that appears without any corresponding perturbation in host-side metrics, is precisely the kind of pattern that a system monitoring only one plane is structurally incapable of perceiving.

Table 4.2 lists each feature alongside its detection rationale and the STRIDE categories to which it primarily contributes.

Table 4.2: Feature Rationale and STRIDE Attack Indications

Feature	**Description and Security Rationale**
`cpu, memory, disk, load`	Joint resource utilisation indicators whose co-movement patterns distinguish computation-intensive attacks from network-only flooding. Sustained, correlated elevation across these four features is characteristic of resource-exhaustion-based **Denial of Service**; isolated changes in disk and process-level indicators point toward **Tampering**.

Feature	Description and Security Rationale
`uptime`	A stability counter whose value decrements only when the device restarts. Unscheduled reductions indicate either crash-induced reboot under load or a deliberate reset used to activate a planted persistence mechanism. Relevant to both **Denial of Service** and **Tampering**.
`bytes_sent`, `bytes_recv`	Interface throughput pair. Asymmetric outbound elevation against a stable inbound baseline is characteristic of data exfiltration (**Information Disclosure**); symmetric high-volume elevation of both counters points toward network flooding (**Denial of Service**).
`tcp_conn`, `open_ports_count`	Session and service-exposure indicators. Rapid growth in active connections suggests scanning or lateral movement; new ports absent from the baseline service inventory indicate pre-exfiltration staging. Both map primarily to **Spoofing** and **Information Disclosure**.
`icmp_packets`	ICMP traffic volume. Counts substantially above the device baseline are consistent with host-discovery sweeps or participation in a network-layer flood. Contributes to detection of **Information Disclosure** and **Denial of Service**.
`latency`	End-to-end round-trip time serving as a compound signal: elevated values may reflect external network congestion from flooding or internal processing delay propagating through to response behaviour. Provides supplementary evidence across multiple STRIDE categories, most directly **Denial of Service**.

Feature	Description and Security Rationale
`tamper_file_check`	Host-level integrity indicator derived from periodic verification of critical system files. Any discrepancy constitutes direct evidence of **Tampering** and, in multi-stage intrusion sequences, may precede a subsequent **Information Disclosure** event by a measurable interval.

4.4 Statistical Behavioural Profiling Model

The profiling model rests on a deliberate choice of simplicity over expressiveness. Rather than estimating a complex generative model of device behaviour, it characterises each feature with two summary statistics derived from observations recorded during confirmed benign operation. This choice was made on two grounds. First, the model must be interpretable to practitioners who need to trace an alert back to a specific measurement deviation; a model whose internal representation is opaque provides no actionable diagnostic output. Second, the model runs on hardware at the constrained end of the residential IoT spectrum and must impose computational overhead that is, in practical terms, negligible. Gaussian summary statistics satisfy both requirements and, as the validation experiment in Section 5.2.1 demonstrates, produce substantially better detection outcomes than the global-threshold alternative.

For device d and feature f_j, measured across n benign observations, the sample mean is:

$$\mu_j = \frac{1}{n} \sum_{i=1}^{n} f_j(i) \tag{4.1}$$

and the sample standard deviation is:

$$\sigma_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\big(f_j(i) - \mu_j\big)^2} \tag{4.2}$$

The pair (μ_j, σ_j) defines a symmetric envelope within which normal operation is expected to fall:

$$f_j^{\text{normal}} \in \big[\mu_j - k\sigma_j,\ \mu_j + k\sigma_j\big] \tag{4.3}$$

The width of this envelope is governed by k. At $k = 2$, approximately 95 % of observations from a Gaussian distribution fall within the bounds; at $k = 3$, the figure rises to 99.7 %. For a continuously running security monitor, where every false positive demands operator attention, $k = 3$ is the pragmatic choice. These parameters feed directly into the normalised z-score computation in Section 4.7 and into the posture deviation model of Section 4.14.

Short-duration measurement artefacts, a transient CPU spike during a scheduled synchronisation, a momentary latency excursion during a connection reset, should not trigger alerts. A sliding-window average applied before the envelope comparison attenuates these single-sample events without blunting sensitivity to sustained anomalies:

$$\bar{f}_j(t) = \frac{1}{|W|}\sum_{t' \in W} f_j(t') \tag{4.4}$$

The complete profiling output for device d is a vector of per-feature parameter pairs:

$$P_d = \big\{(\mu_j,\, \sigma_j)\big\}_{j=1}^{7} \tag{4.5}$$

On the Gaussian Assumption

Adopting the symmetric interval $[\mu_j - k\sigma_j, \mu_j + k\sigma_j]$ should not be read as a claim that every monitored feature follows a normal distribution in practice. Several features, `cpu` and `icmp_packets` notably, are positively skewed under benign conditions, as the asymmetric standard deviations in Table 5.1 confirm. Three aspects of the design limit the practical consequence of this approximation. The downstream Random Forest classifier operates on the normalised deviation index $A_i(t)$ rather than raw feature values, and as a non-parametric model it learns decision boundaries without requiring any distributional assumption on its inputs, thereby absorbing distortion introduced by skewed marginals. The sliding-window average of Equation (4.4) reduces moment-to-moment skew before any threshold comparison is made. And fitting separate profiles per device, rather than a single model across the population, accommodates the fact that devices occupying different roles in the network produce characteristically different distributional shapes; pooling them would force a single envelope to represent behaviour that is genuinely bimodal at the population level.

4.5 Profiling-Grounded Anomaly Detection

Statistical characterisation of normal device behaviour is a prerequisite for detection, not detection itself. The per-device Gaussian envelopes constructed in Section 4.4 describe the expected operating range of each node across eleven Sys-Net features; what they do not do, standing alone, is evaluate incoming observations against that range and act on the result. Bridging that gap requires a monitoring mechanism capable of running continuously, scoring incoming telemetry against the learnt baseline, and escalating statistically significant departures to the downstream classification layer.

Meeting that requirement without introducing the practical vulnerabilities of conventional detection approaches poses two distinct challenges. Signature-based detection

methods fail silently against threats outside their catalogue, and fixed global thresholds calibrated for one device population are reliably miscalibrated for another. The approach developed here anchors every detection criterion directly in the per-device baseline: an observation triggers a candidate anomaly signal not because it crosses some population-level constant but because it falls at a distance from that specific device's learnt distribution that benign operation alone cannot plausibly explain. Grounding the detection mechanism in individually learnt profiles gives the framework a degree of generalisability to unseen attack types that signature-based systems cannot achieve, because the operative question is always whether observed behaviour is consistent with the established normal, rather than whether it matches a catalogued threat pattern.

A further design decision concerns the form of the detection output. Conventional detectors collapse the detection decision to a binary alarm, discarding all evidence about which features drove the result and by how much. That evidence is precisely what the STRIDE-aligned classification layer in Section 4.10 requires to make semantically meaningful threat attributions. Retaining feature-level deviation magnitudes throughout the detection pipeline, rather than replacing them with a flag, preserves the interpretive chain that connects a raw measurement departure to a structured description of probable attacker intent. Both decisions , device-specific grounding and graded output , shape every aspect of the model developed in the sections that follow.

4.6 Behavioural Deviation Detection Framework

The detection layer operates as the real-time monitoring engine of the overall CSA framework, continuously comparing incoming device telemetry against the per-device behavioural profile produced by the profiling module. Its specific mandate within the research programme is to address the following question:

RQ3: To what extent does profiling-grounded anomaly detection improve be-

havioural deviation identification relative to single-modality (device-only or network-only) approaches?

Answering that question credibly required resolving several architectural tensions, each of which produced a concrete design commitment. System-level and network-level measurements are pooled into a single composite anomaly score rather than evaluated in separate detection channels, a decision that enables cross-plane correlation between signals that many real-world IoT attacks distribute across both layers while remaining individually below threshold in either one. Temporal sensitivity is maintained by computing the score over a sliding window of length τ: isolated operational spikes do not trigger alerts, but sustained departures accumulate into a persistent signal that crosses the reporting threshold. Feature-level deviation magnitudes are preserved in the output rather than collapsed into a binary flag, ensuring that the STRIDE-aligned classification layer of Section 4.10 receives structured evidence rather than an undifferentiated alarm.

4.7 Anomaly Detection Model

4.7.1 Statistical Foundation

Consider a device d monitored across m features $\{f_1, f_2, \ldots, f_m\}$. The profiling phase characterised benign behaviour for each feature through a Gaussian model:

$$f_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{4.6}$$

When an incoming observation $f_i(t)$ arrives at time t, it is expressed as a normalised distance from the learnt mean by computing its absolute z-score against the stored profile parameters. Assembling these per-feature distances into a vector yields the Deviation Feature Vector:

$$V_D(t) = [A_1(t),\, A_2(t),\, \ldots,\, A_m(t)], \qquad \text{where} \quad A_i(t) = \frac{|f_i(t) - \mu_i|}{\sigma_i} \tag{4.7}$$

The z-score normalisation accomplishes two things simultaneously. It renders commensurable a set of features whose raw units and scales are entirely heterogeneous , CPU utilisation expressed as a percentage sits alongside ICMP packet count expressed as an integer , by reducing each to a distance measured in standard deviations of its own distribution. It also transforms the feature space so that every element of $V_D(t)$ directly encodes how unexpected the observation is relative to the established profile, rather than simply how large it is. Machine learning applied downstream to this vector therefore operates on evidence of anomaly magnitude, not on raw telemetry whose scale variation would otherwise confound any learning algorithm.

4.7.2 Composite Anomaly Score

A scalar anomaly score aggregating all monitored features is obtained as a weighted sum:

$$A(t) = \sum_{i=1}^{m} w_i\, A_i(t) \tag{4.8}$$

Uniform weights $w_i = 1/m$ are adopted in the baseline implementation, treating every Sys-Net channel as equally informative about behavioural deviation. This is a deliberate simplification rather than an oversight. The downstream Random Forest classifier recovers non-uniform feature importance implicitly through its splitting criterion, which at every tree node selects the feature that maximally separates the classes present in that partition. Keeping the anomaly module agnostic to feature importance therefore preserves its transparency and stability, while leaving the adaptive weighting task to the classification layer where it can be learnt from labelled data rather than specified by

hand.

4.7.3 Threshold and Temporal Filtering

The binary detection decision is governed by a significance threshold θ:

$$A(t) > \theta \;\Rightarrow\; \text{behaviour is anomalous at time } t \tag{4.9}$$

The per-feature profiling envelope uses $k = 3$, corresponding to $\mu_i \pm 3\sigma_i$ and approximately 99.7 % coverage under the Gaussian assumption. Because the deviation indices $A_i(t)$ are normalised by σ_i, the anomaly score $A(t)$ is dimensionless; θ therefore represents a threshold in normalised-deviation units rather than a raw $\mu \pm k\sigma$ boundary. Two mechanisms address the well-known limitations of any fixed threshold. The unsupervised models evaluated in Section 4.8 learn adaptive decision boundaries from training data, offering an alternative detection criterion that does not depend on the Gaussian assumption holding exactly. Dynamic profiling periodically recalculates (μ_i, σ_i) from a background sliding window of recent benign traffic, allowing the baseline to respond gradually to genuine long-term changes in device behaviour without being corrupted by sustained adversarial activity that would otherwise drag the mean toward the attack distribution.

Transient operational fluctuations , a brief CPU spike during a scheduled backup, a momentary latency excursion during a connection reset , should not trigger alerts. A temporal persistence requirement prevents this by reporting an anomaly only when the fraction of time steps within the preceding window of length τ that exceeded θ surpasses a coverage proportion α:

$$\frac{1}{\tau} \sum_{t' \in [t-\tau,\, t]} \mathbf{1}_{A(t') > \theta} \;>\; \alpha \tag{4.10}$$

Once a persistent anomaly is confirmed, the full Deviation Feature Vector $V_D(t)$ is forwarded to the STRIDE classification layer. Forwarding the complete vector rather than a summary flag preserves the per-feature evidence that drove the detection decision, maintaining the interpretive chain that connects a raw measurement deviation to a structured threat attribution at every subsequent stage of the pipeline.

4.8 Machine Learning Models for Anomaly Detection

Four unsupervised algorithms were evaluated alongside the primary profiling-grounded mechanism. Bringing in multiple methods was not intended to identify a replacement for the statistical baseline approach; the purpose was to interrogate the properties of the hybrid Sys-Net dataset by observing how methods with qualitatively different inductive biases respond to the same data. How each method fails, and where, discloses something about the underlying structure of benign and attack traffic in the cross-layer feature space that no single algorithm evaluation could reveal.

Unsupervised methods were specifically chosen over supervised alternatives because of a practical constraint that characterises real IoT security deployments. A comprehensive labelled corpus covering every plausible attack variant does not exist for any residential IoT environment, and the continuous evolution of adversarial techniques means it could not be compiled once and held fixed. An unsupervised approach learns the multivariate shape of benign behaviour and treats substantial departures from it as suspicious, without requiring attack-class labels at training time. Viewed this way, the four methods evaluated here are complementary to the supervised Random Forest classifier in Section 4.10 rather than competitive with it.

4.8.1 Unsupervised Anomaly Detection Algorithms

k-Nearest Neighbours and Local Outlier Factor. Both kNN and LOF locate anomalies through proximity relationships, but at different scales of locality. kNN scores each sample by the distance to its k-th nearest neighbour, so observations far from any benign cluster receive high scores regardless of local density variation. LOF extends this by computing the ratio of a point's local reachability density to that of its neighbours, surfacing anomalies embedded within otherwise compact behavioural regions where global distance measures may be misleading. On high-dimensional hybrid telemetry, both methods are susceptible to the curse of dimensionality, and their performance is expected to degrade wherever benign and attack distributions overlap substantially in the feature space.

Histogram-Based Outlier Score. HBOS estimates the univariate distribution of each feature independently via a histogram, then scores observations by the product of their inverse-probability contributions across all features. Its principal advantage is computational efficiency: estimation scales linearly with the number of features and samples, making it attractive for resource-constrained deployments. The cost of that efficiency is the feature-independence assumption, which means HBOS cannot detect anomalies arising from unusual combinations of individually plausible values. In a hybrid telemetric setting where inter-feature correlations carry much of the discriminative signal, this limitation is consequential.

Isolation Forest. Rather than measuring proximity to the normal cluster, Isolation Forest assesses how easily a point can be separated from the rest of the dataset. An ensemble of randomly constructed decision trees recursively partitions the feature space; points in sparse, atypical regions require fewer splits to isolate and thus receive short average path lengths, which translate into elevated anomaly scores. The mechanism is distribution-agnostic, scales well to high-dimensional data, and requires only a single

tunable hyperparameter , the contamination fraction.

Within the adversarial robustness evaluation of Section 4.14, Isolation Forest assumes a structurally distinct role as a per-class confirmation gate. Two dedicated forests are instantiated for each STRIDE attack class: a source-class forest (IF_{src}) fitted on unperturbed feature vectors representing class c, and a normal-class forest (IF_{norm}) fitted exclusively on clean benign traffic. A perturbed sample that successfully shifts its feature distribution toward the benign cluster may nonetheless fail to reproduce the full multivariate structure of genuine benign traffic; IF_{norm} detects precisely this discrepancy. Contamination parameters are calibrated per class to reflect empirically observed outlier proportions: $c_{\text{Normal}} = 0.05$; $c_{\text{DoS}} = c_{\text{Spoofing}} = c_{\text{Info}} = 0.10$; $c_{\text{Tampering}} = 0.15$.

Adversarial Application: Per-Class Confirmation Gates. Within the adversarial robustness framework of Chapter 5, the two Isolation Forest instances enter the master CSA detection gate as binary anomaly indicators. A prediction of -1 from either forest signals that the sample does not occupy its claimed class neighbourhood , confirming that the sample has been displaced from its original class distribution, from the benign distribution, or from both. Both forests are fitted on the augmented training partition with the contamination parameters specified above. The formal structure of the CSA gate is presented in Section 4.14.

4.9 STRIDE-Aligned Intrusion Detection

Knowing that a device has departed from normal operation is not, by itself, sufficient for a security analyst who must decide how to respond. Anomaly detection marks the departure; what remains undetermined is its character, whether the deviation reflects an opportunistic probe, an attempt to tamper with system state, a deliberate effort to exhaust resources, or something altogether benign. Closing that interpretive gap is the motivation for this chapter.

The approach taken here builds on the profiling and anomaly detection layers described in Chapter 3 and Section 4.7 by introducing a dedicated intrusion detection layer aligned with the STRIDE threat taxonomy (Shevchenko et al., 2018). Rather than treating every anomaly as an undifferentiated alarm, the proposed layer maps observed deviations onto one of six canonical threat categories, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, or Elevation of Privilege, thereby associating low-level measurement data with structured descriptions of attacker intent. The six-category formulation represents the general STRIDE-aligned architecture. The empirical instantiation evaluated in this thesis is restricted to the four categories observable in SmartHome-StrideDS: Spoofing, Tampering, Information Disclosure, and Denial of Service. Repudiation and Elevation of Privilege remain architecturally representable but were not empirically implemented or evaluated in the present study. The result is a detection output that can be acted upon meaningfully by operators and fed, at a higher level of abstraction, into downstream risk quantification processes.

4.10 IDS Architecture

Four interdependent functional components together constitute the proposed IDS layer. Each component addresses a distinct aspect of the translation from raw telemetry to structured security intelligence.

Anomaly Interpretation. When an upstream alert is issued, the IDS re-examines the feature-level data that triggered it. Real-time telemetry vectors are compared against the device-specific behavioural profiles constructed during the profiling phase, and the resulting deviation magnitudes are assembled into a normalised anomaly vector $\mathbf{a}_d(t)$. This step is necessary because anomaly detectors operate on aggregate signals; the IDS, in contrast, must resolve which individual features are driving the departure and to what degree.

Threat Classification. A multi-class machine-learned classifier receives the feature-level evidence for observations escalated by the profiling-based anomaly detector and assigns each observation to a supported STRIDE category. The classifier was trained on SmartHome-StrideDS, a hybrid Sys-Net dataset whose labels are aligned with STRIDE semantics. The training data associate certain feature patterns with particular threat categories. For example, sustained increases across both `cpu` and system `load` are associated with Denial of Service observations; abrupt expansion of `open_ports_count` is associated with reconnaissance observations labelled as Information Disclosure; and failures surfaced through `tamper_file_check` validation are associated with Tampering-labelled observations. These feature–threat associations provide evidence for classification within the available dataset, but are not uniquely causal indicators of attacker intent.

Context-Aware Security Adaptation. A design decision that distinguishes this IDS from threshold-only approaches is its incorporation of device-specific operational context. The same value of `bytes_sent` may be entirely unremarkable for a high-throughput surveillance camera while warranting urgent scrutiny for a thermostat whose normal outbound traffic is minimal. Beyond device type, the vulnerability intelligence generated during the profiling phase (Chapter 3) can be used to adjust detection sensitivity dynamically: a device carrying known unpatched CVEs may receive tightened thresholds so that borderline deviations are escalated rather than suppressed.

Response Signalling. Following classification, the IDS emits structured alerts to the central monitoring dashboard, carrying the identified STRIDE category, the computed threat score, and device-level impact metadata. Automated mitigation is not yet implemented and remains an avenue for future work; the present architecture is designed as an observation and signalling layer whose outputs support human analysts and feed downstream risk quantification.

4.11 STRIDE Threat Categorisation

The mapping from measured deviations to STRIDE threat scores follows a weighted linear aggregation scheme, formalised in Algorithm 1. The procedure admits a compact mathematical description while remaining transparent enough for security practitioners to inspect and calibrate.

Algorithm 1: STRIDE Threat Scoring and Categorisation

Input: Anomaly Vector $\mathbf{a}_d(t) = [A_1, A_2, \ldots, A_m]$;
Mapping Matrix $R \in \mathbb{R}^{m \times 6}$, $r_{ij} \in [0, 1]$;
Sensitivity Thresholds $\Phi = \{\phi_1, \ldots, \phi_6\}$
Output: Set of active STRIDE Alerts $\mathcal{A}$

Initialization: $\mathcal{A} \leftarrow \emptyset$;
for *each STRIDE category* $j \in \{1, \ldots, 6\}$ **do**
    `// Step 1: accumulate weighted score`
    $S_j(t) \leftarrow 0$;
    for *each profiling feature* $i \in \{1, \ldots, m\}$ **do**
        $S_j(t) \leftarrow S_j(t) + \big(r_{ij} \times A_i(t)\big)$;
    `// Step 2: context-aware thresholding`
    if $S_j(t) > \phi_j$ **then**
        Generate alert for category j;
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\text{Category}_j,\ S_j(t)\}$;
return $\mathcal{A}$;

Algorithm 1 is formulated over the entire six-category STRIDE taxonomy for architectural completeness. In the experiments reported in Chapter 5, only the four empirically observable categories—Spoofing, Tampering, Information Disclosure, and Denial of Service—were instantiated and evaluated.

4.11.1 Formal Specification

Let $\mathbf{a}_d(t) = [A_1(t), A_2(t), \ldots, A_m(t)]$ denote the anomaly vector for device d at time t, where each entry $A_i(t)$ records the normalised deviation of profiling feature f_i from its established baseline. Values exceeding unity indicate departures greater than one standard deviation. For the eleven-feature SmartHome-StrideDS configuration, this vector is

written explicitly as:

$$\mathbf{a}_d(t) = [\, A_{\texttt{cpu}}(t),\ A_{\texttt{memory}}(t),\ A_{\texttt{disk}}(t),\ A_{\texttt{load}}(t),\ A_{\texttt{uptime}}(t),$$
$$A_{\texttt{bytes_sent}}(t),\ A_{\texttt{bytes_recv}}(t),\ A_{\texttt{tcp_conn}}(t),\ A_{\texttt{icmp_packets}}(t),\ A_{\texttt{open_ports_count}}(t),\ A_{\texttt{latency}}(t)\,]$$

These eleven positions correspond, in order, to: `cpu`, `memory`, `disk`, `load`, `uptime`, `bytes_sent`, `bytes_recv`, `tcp_conn`, `icmp_packets`, `open_ports_count`, and `latency`.

A mapping matrix $R \in \mathbb{R}^{m\times 6}$ encodes the relationship between each feature and the six STRIDE categories (S: Spoofing, T: Tampering, R: Repudiation, I: Information Disclosure, D: Denial of Service, E: Elevation of Privilege):

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{16} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{m6} \end{bmatrix}, \qquad r_{ij} \in [0,1]$$

Each coefficient r_{ij} quantifies the degree to which feature f_i is associated with threat class j. All six columns remain present in the formal architecture; however, only the columns corresponding to Spoofing, Tampering, Information Disclosure, and Denial of Service were empirically implemented and parameterised using SmartHome-StrideDS in this thesis. The Repudiation and Elevation of Privilege columns were not empirically parameterised or evaluated. Three indicative assignments for the implemented categories are:

$$r_{\texttt{open_ports_count},\,\text{Info. Disc.}} = 0.9, \qquad r_{\texttt{tamper_file_check},\,\text{Tampering}} = 1.0, \qquad r_{\texttt{cpu},\,\text{DoS}} = 0.8$$

The STRIDE score for category j at time t is then:

$$S_j(t) = \sum_{i=1}^{m} r_{ij} \cdot A_i(t) \tag{4.11}$$

A category is flagged when $S_j(t)$ surpasses the category-specific sensitivity threshold:

$$S_j(t) > \phi_j \tag{4.12}$$

The threshold ϕ_j is not fixed globally; it is calibrated to reflect the device's operational context, its typical traffic patterns, and its known vulnerability status. Devices carrying unpatched CVEs attract a reduced ϕ_j, heightening sensitivity to marginal deviations. Conversely, a device under legitimately elevated computational load receives an increased ϕ_j to prevent operational fluctuations from generating spurious alerts.

4.11.2 Feature–Intent Relationships

The mapping matrix encodes four principal associations between observable features and STRIDE threat classes. Spoofing is suggested by unexpected identity or session anomalies and abnormal changes in open port activity that may indicate an impersonation attempt. Tampering manifests through irregularities in file integrity metrics and measurable configuration drift between successive profiling windows. Information Disclosure is associated with elevated network throughput, outbound connections to unexpected endpoints, and traffic patterns consistent with pre-exfiltration staging. Denial of Service is reflected in sustained overload of CPU and memory resources, abnormal ICMP volumes, and conditions that degrade or interrupt service availability. These feature–threat relationships are associations used to interpret classification outcomes, not uniquely causal relationships: the same indicators may arise from benign activity or from different threat scenarios. The assigned category is therefore interpreted within the labelled training context and is not a direct consequence of STRIDE itself.

4.12 Continuous Security Assurance Integration

Security monitoring systems tend to be built around the question of *whether* something is wrong. The preceding sections have answered that question with considerable precision: Section 4.4 established what normal looks like on a per-device basis; Section 4.7 trained a classifier to recognise departures from that norm; Section 4.10 extended the analysis to attribute each departure to a category of attacker intent drawn from the STRIDE taxonomy. Taken together, these contributions form a detection pipeline of real capability. Yet capability is not the same as operational utility, and the distinction matters here in a particular way.

An operator monitoring a smart-home network during an active shift does not, in practice, want to be told that something anomalous has occurred on IoT-04 and that it has been classified as a Denial of Service event. That information is necessary but not sufficient for decision-making; what the operator actually needs is a sense of the current state of the environment as a whole, expressed as a single quantity that rises and falls in step with the real-world threat level and that can be acted upon without first consulting the underlying telemetry. No such quantity emerges from an episodic detection pipeline. The pipeline emits a stream of events; it does not maintain a running account of how bad things currently are.

Continuous Security Assurance, referred to throughout as CSA, was built to close that gap. The framework assigns a single numerical posture score to the monitored network, refreshes that score continuously as new telemetry arrives, and ensures that the evidence underlying the score is visible to the operator alongside the number itself. The score is not a binary: it is a graded, dynamic quantity that climbs toward nominal as anomalies resolve and falls in proportion to both the number of concurrently affected devices and the severity of each individual deviation. The design principle is that an operator should be able to read the state of the network as readily as one reads a dashboard instrument, without specialist training in the interpretation of security alerts.

Section 4.13 describes the architecture of the CSA layer and situates it within the four-stage processing hierarchy. The three consistency verification rules that protect the framework against adversarially crafted inputs, together with the Isolation Forest gates that reinforce them, are developed in Section 4.14. Section 4.15 derives the mathematical model that maps per-device feature deviations onto the aggregate posture score. The operator dashboard is described in Section 4.16. Section 5.6 subjects the complete framework to three operationally distinct test conditions; and Section 4.17 examines end-to-end pipeline behaviour alongside the vulnerability characterisation methodology. Chapter 6 draws these contributions together.

4.13 CSA Framework Integration

The CSA layer is not a standalone component that happens to receive inputs from the detection pipeline. It is the apex of a four-stage hierarchy in which each lower stage performs a well-defined transformation on the telemetry and passes the result upward, so that the posture score ultimately reflects the accumulated evidence from every layer of the architecture. The contribution of each stage to that accumulation is described in turn.

Stage 1, behavioural profiling. The identification of anomalies presupposes a model of normality, and that model must be specific to each individual physical device rather than pooled across the device population, since different devices in a smart-home network exhibit markedly different baseline behaviour. During a training window of unattacked operation, two statistics are therefore recorded for every monitored feature j on each device: the mean μ_j and the standard deviation σ_j. These statistics, taken jointly across the full feature set for a given device, constitute the benign baseline against which subsequent observations are compared. There is no shortcut to this step; without it, no subsequent stage has a reference point from which to measure departure.

Stage 2, anomaly detection. At each sampling instant t, the Sys-Net telemetry pipeline emits an n-dimensional feature vector,

$$X(t) = \big\{F_1(t),\ F_2(t),\ \ldots,\ F_n(t)\big\}, \tag{4.13}$$

in which each entry corresponds to one monitored metric. A trained Random Forest classifier, designated $\mathrm{RF}_{\mathrm{AD}}$, reads this vector and returns a binary label: anomalous or not. The telemetry feeding into $\mathrm{RF}_{\mathrm{AD}}$ is hybrid throughout, fusing operating-system-level signals with network-layer signals end-to-end. Earlier chapters of this thesis established empirically that the hybrid combination outperforms either single-layer alternative on both precision and recall, and there was no reason to depart from that choice at this stage of the pipeline.

Stage 3, STRIDE classification. Samples flagged as anomalous by Stage 2 are forwarded to a second classifier, $\mathrm{RF}_{\mathrm{IDS}}$, whose function is to assign each one to a category of attacker intent drawn from a four-class STRIDE-derived taxonomy: Spoofing (S), Denial of Service (D), Tampering (T), and Information Disclosure (I). The operational importance of this step is less about classification accuracy in the abstract than about the qualitative difference between attack categories in their consequences and remediation requirements. An operator who knows only that an anomaly has occurred cannot formulate a targeted response; an operator who knows that the anomaly is a Spoofing event on a specific device can.

Stage 4, posture quantification. The attack labels issued by Stage 3 and the per-feature deviation magnitudes computed by Stage 2 together enter the posture model described in Section 4.15. That model returns the real-valued score $\mathrm{CSA}(t) \in (0,\ 100]$, which reaches the dashboard alongside the STRIDE label and a breakdown of the features that contributed most to the current score value.

4.14 CSA Consistency Verification Layer

The vulnerability of a trained classifier to adversarially crafted inputs is not eliminated by adversarial training; it is, at best, made marginally more expensive to exploit. The decision boundary shifts, but the structural exposure remains. The CSA layer responds to this limitation not by attempting to train it away but by interposing a separate verification stage between the classifier's output and any downstream action. Every incoming sample is subjected to three checks, each grounded in a property, either statistical or causal, that genuine network traffic must satisfy. A sample that clears the Random Forest but trips one of these checks is blocked before it reaches the posture model. Two Isolation Forest gates constitute an additional line of defence, and their conjunction with the rule-based checks forms the master decision gate formalised in Section 4.14.5.

4.14.1 Rule 1: Temporal Statistical Consistency

The defining statistical expectation of membership in an attack class is that genuine samples drawn from that class should resemble other members of it. This is not a demanding requirement: it does not require that samples be typical, only that they not be outliers in a strong sense. Adversarially perturbed samples frequently fail even this weak test, because the perturbation concentrates on whichever features most influence the classifier's decision and pushes them far enough from the class centre to render them statistical anomalies. Rule 1 quantifies this displacement. For each feature i, the standardised distance between the perturbed value $f_i(x')$ and the class mean μ_i is computed as

$$Z_i = \frac{\left| f_i(x') - \mu_i \right|}{\sigma_i}, \tag{4.14}$$

where σ_i denotes the within-class standard deviation for feature i. A value of Z_i exceeding 3.0 locates the perturbed observation more than three standard deviations from its

class mean, an event that occurs infrequently in authentic traffic and with near-routine frequency in adversarial samples. The rule triggers when a majority of features exceed this threshold simultaneously,

$$T_{\text{EWMA}} \text{ fails} \iff \sum_{i=1}^{N} \mathbf{1}(Z_i > 3.0) \geq \frac{N}{2}. \tag{4.15}$$

The majority criterion rather than a universal-quantifier threshold was chosen because adversarial perturbations are characteristically concentrated, not diffuse: the adversary perturbs the subset of features that maximises the probability of a label flip and leaves the remainder largely undisturbed. A rule requiring every feature to be displaced would therefore miss the majority of adversarial samples, while a rule requiring only one would be hopelessly sensitive to measurement noise. Empirical calibration confirmed that requiring displacement in just over half the features strikes the right balance between catching adversarial inputs and avoiding false alarms on benign traffic; tighter thresholds improved attack detection at the cost of false-positive rates that undermined operator trust.

4.14.2 Rule 2: Physical Causal Consistency

Network behaviour exhibits not merely statistical regularities but causal ones, rooted in the underlying mechanics of packet transmission and resource allocation. When a denial-of-service flood begins, connection count and latency increase together because the same flood event drives both; when a cryptographic side-channel exploit is in progress, CPU load and memory pressure co-move because both are consequences of the same underlying computation. An adversary who perturbs one member of a causally coupled pair while leaving the other undisturbed produces a combination that no genuine traffic event would generate, and Rule 2 is designed to identify exactly this signature.

The signed change in each feature between the original and the perturbed sample

is expressed as

$$\Delta_A = f_A(x') - f_A(x), \qquad \Delta_B = f_B(x') - f_B(x), \tag{4.16}$$

and a causal violation is recorded whenever feature A increases while its coupled partner B does not follow,

$$\text{Violation} = \mathbf{1}\big(\Delta_A \uparrow \text{ but } \Delta_B \not\uparrow\big). \tag{4.17}$$

The rule C_{corr} blocks the sample once three or more violations have accumulated across the encoded causal pairs. A threshold of one would make the rule unusably brittle: ordinary measurement noise produces occasional apparent mismatches between coupled features even in clean traffic, and a single violation would generate unacceptable false-positive rates. Three violations provide sufficient tolerance for noise while still catching the coordinated, multi-feature manipulation that any adversarial perturbation serious enough to flip a classifier label must entail.

4.14.3 Rule 3: STRIDE Semantic Viability

Each category in the STRIDE taxonomy is defined, in part, by the operational footprint it must leave in the network. A denial-of-service attack cannot be a denial-of-service attack without elevated traffic volumes; a spoofing attack cannot be a spoofing attack without forged identity fields at anomalous rates. These are not statistical tendencies that an attacker might opt out of; they are constitutive features of what it means for an event of each type to be occurring at all. Rule 3 exploits this constitutive relationship. An adversary who suppresses the signature features of an attack class in order to evade detection has not merely changed how the attack looks; they have changed what the attack is, and in the limit they have dismantled it entirely. A sample with no surviving

signature features is not a cleverly disguised attack; it is simply not an attack.

For a sample predicted to belong to attack class c, the count of signature features whose values remain at least 50 % above the class mean after perturbation is

$$E = \sum_{i \in \mathcal{F}^c} \mathbf{1}\big(f_i(x') \geq 1.5\,\mu_i\big), \tag{4.18}$$

where $\mathcal{F}^c$ is the set of signature features for class c. The gate blocks the sample when $E = 0$: a sample from which every signature feature has been suppressed below the viability threshold is semantically incompatible with any genuine instantiation of the attack class it has been assigned to, and blocking it requires no assumption about the adversary's intent beyond the observation that the sample cannot correspond to a real attack.

4.14.4 Isolation Forest Confirmation Gates

There remains a residual category of adversarial input: samples that satisfy all three consistency rules yet inhabit a region of feature space that is, on any reasonable account, improbable. Such samples neither resemble authentic members of the attack class they have been assigned to nor resemble clean normal traffic; they occupy the void between the two known distributions. Two Isolation Forests are deployed to detect occupancy of precisely this void. Each returns -1 for an anomaly and $+1$ for a normal sample,

$$\mathrm{Anom}_{\mathrm{src}} = \mathbf{1}\big(\mathrm{IF}_{\mathrm{src}}(x') = -1\big), \qquad \mathrm{Anom}_{\mathrm{norm}} = \mathbf{1}\big(\mathrm{IF}_{\mathrm{norm}}(x') = -1\big). \tag{4.19}$$

The two forests differ in what they regard as normal. $\mathrm{IF}_{\mathrm{src}}$ is trained on labelled samples from the source attack class and is therefore sensitive to departures from the

statistical structure of that class. $\mathrm{IF}_{\mathrm{norm}}$ is trained on clean normal traffic and is sensitive to departures from that distribution instead. A sample that neither resembles attack traffic nor resembles normal traffic will be flagged by both.

Isolation Forest was preferred over distance-based alternatives for several reasons, of which the most important is geometric rather than computational. The anomaly score produced by an Isolation Forest is a function of path length through randomly constructed partition trees: samples in sparse, low-density regions of feature space are isolated in few partitioning steps, receiving short paths and correspondingly high anomaly scores. Adversarial perturbations almost invariably push samples into exactly those sparse inter-class regions, and path-length isolation is therefore intrinsically well suited to the adversarial detection task, even when rule-based checks have been cleared. The computational considerations reinforce this preference: distance-based methods such as k-nearest neighbours and Local Outlier Factor incur pairwise computation costs that scale poorly with dataset size, and under experimental conditions on the TON Linux process dataset both recorded denial-of-service recall values below 16 %, whereas Isolation Forest sustained substantially higher recall under identical conditions. From a practical tuning standpoint, Isolation Forest exposes a single hyperparameter, the contamination fraction, while One-Class SVM and Local Outlier Factor together require several kernel and neighbourhood parameters to be tuned jointly.

4.14.5 The Master Decision Gate

The three consistency rules and the two Isolation Forest gates are combined into a single blocking predicate. A sample is blocked when at least one of the following three conditions holds,

$$\begin{aligned} \text{Block} \;=\; & \underbrace{(\text{physical checks fail} \;\wedge\; \text{Anom}_{\text{src}})}_{\text{Condition 1}} \\ & \vee\; \underbrace{(\text{physical checks fail} \;\wedge\; \text{Anom}_{\text{norm}})}_{\text{Condition 2}} \\ & \vee\; \underbrace{(\text{Anom}_{\text{src}} \;\wedge\; \text{Anom}_{\text{norm}})}_{\text{Condition 3}}, \end{aligned} \tag{4.20}$$

where the shorthand "physical checks fail" denotes the disjunction of the three rule failures,

$$\text{physical checks fail} \;=\; \neg T_{\text{EWMA}} \;\vee\; \neg C_{\text{corr}} \;\vee\; \neg S_{\text{STRIDE}}. \tag{4.21}$$

Each condition is directed at a qualitatively distinct evasion strategy and covers a region of the adversarial input space that the other two do not. Condition 1 fires when a consistency rule has raised a flag and the sample additionally fails to resemble any known member of its claimed attack class. Condition 2 fires when a consistency rule has triggered and the sample further bears no statistical resemblance to clean normal traffic; an attacker who attempted to evade this condition by mimicking benign traffic would need to reproduce the full joint multivariate distribution of genuine normal operation from within the constraints of a bounded perturbation budget, a requirement that is for practical purposes unattainable. Condition 3 is the most architecturally significant of the three, because it requires no rule to have fired at all: it catches the adversary who has successfully navigated every rule-based check but in doing so has necessarily deposited the perturbed sample in the geometric dead space between the two known distributions, where both Isolation Forests independently register its improbability.

The three conditions in conjunction impose on any adversary a constraint satisfaction problem that is, within a realistic L^{∞} budget, infeasible. A successful evasion would require all of the following to hold simultaneously: the Random Forest must be fooled; every feature must remain within three standard deviations of its class mean;

every causally coupled feature pair must move in the expected direction; at least one signature feature must remain above the attack-viability threshold; and the full joint distribution of genuine normal traffic must be reproduced. No finite perturbation budget admits a solution satisfying all of these constraints simultaneously.

4.15 Security Posture Quantification Model

Accurate detection is a necessary condition for a useful security monitoring system, but it is not a sufficient one. An operator who receives only event notifications, however precisely classified, is still required to assess the aggregate state of the network from a stream of individual alerts, a cognitive burden that scales badly with the number of devices and the rate of alert generation. The posture model converts the outputs of the detection pipeline into a single continuous score that carries the aggregate assessment the operator needs. It does so through three successive computations.

Per-feature deviation scoring. At each time step, the deviation of feature j from its benign mean is normalised into a dimensionless quantity by dividing by the within-device standard deviation,

$$z_j(t) = \frac{\left|F_j(t) - \mu_j\right|}{\sigma_j + \varepsilon}, \tag{4.22}$$

where $\varepsilon \approx 10^{-6}$ prevents division by zero in the degenerate case of a feature whose training variance is identically zero. Large values of $z_j(t)$ mark features that have moved substantially away from their benign operating range; these are the features that the dashboard subsequently highlights as the most probable contributors to an elevated score.

Weighted risk accumulation. Security relevance is not distributed uniformly across features. A sustained spike in outgoing packet volume is a qualitatively stronger indicator of an active attack than a transient increase in resident memory, and any ag-

gregate that treats the two as equivalent will be misleading. The model captures this asymmetry by multiplying each deviation $z_j(t)$ by a security-relevance weight w_j before summation, and caps the product at $z_{\max}$ to prevent any single catastrophically anomalous feature from overwhelming the contributions of all others,

$$\text{Risk}(t) = \sum_{j=1}^{n} w_j \cdot \min\big(z_j(t),\ z_{\max}\big). \tag{4.23}$$

Both w_j and $z_{\max}$ are calibrated during the profiling phase, drawing on domain knowledge of which features are most diagnostic for each of the four STRIDE categories under evaluation.

Posture score mapping. The accumulated risk is projected onto the final posture score by an exponential decay function,

$$\text{CSA}(t) = 100 \cdot e^{-\text{Risk}(t)}. \tag{4.24}$$

The choice of the exponential form reflects several considerations. The function maps the unbounded accumulator $\text{Risk}(t) \in [0, \infty)$ onto the compact interval $(0,\ 100]$ without requiring any explicit saturation or clipping step, which is both analytically convenient and numerically clean. The calibration properties in the region of operational interest are also well-behaved: a confirmed single-device attack, corresponding to approximately $\text{Risk}(t) = 1.0$, places the score near 37, well below any threshold at which an operator might reasonably withhold action, while a marginal anomaly at $\text{Risk}(t) = 0.1$ yields a score above 90, consistent with a network operating normally. Finally, the exponential form is the conventional choice in the systems dependability literature for expressing reliability and availability (Trivedi, 2016), which places the CSA score in a tradition of measurement that facilitates comparison with related frameworks. The calibration properties described here were verified empirically through the scenario evaluations reported in Section 5.6.

4.16 Real-Time Monitoring Dashboard

The posture score is, by itself, an instrument reading. Its operational value depends entirely on whether it reaches the operator in a form that supports immediate decision-making, which in practice means that both the number and the evidence behind it must be presented together, in real time, and without requiring the operator to interpret technical artefacts. The dashboard was designed with this constraint as its primary specification.

Each of the nine monitored devices is represented as a card in a 3×3 grid. The background colour of each card tracks the per-device posture score: green for values above approximately 85, amber across the intermediate range, and red once the score drops below a configurable alert threshold. The colour scheme allows the operator to read the health of the entire device population without attending to individual numbers as readings arrive, since the most safety-critical transitions, from green to amber and from amber to red, are perceptible in peripheral vision without directed attention.

Each card carries two further elements beyond the colour indicator. The first is the STRIDE category of the most recently confirmed detection on that device, displayed as a label whenever a confirmed event is in progress. This matters because the remediation response to a Spoofing event differs substantially from the response to a Denial of Service: knowing the category is what allows the operator to dispatch the appropriate response rather than a generic one. The second element is a list of the Sys-Net features that contributed most to the current risk score. An operator who observes `tcp_conn` and `pkt_rate` flagged in combination on a device card can conclude with reasonable confidence that the anomaly originates in network-layer flooding rather than a process-level intrusion, without consulting any further source. This feature-level transparency also serves investigators conducting post-incident analysis, since it provides a documented and timestamped trace of which metrics deviated, and by how much, throughout the duration of the event.

The score computation runs continuously; there is no polling schedule that would

introduce latency between sample arrival and display update. As each new observation arrives from the Sys-Net pipeline, the risk accumulator recalculates and the displayed scores refresh immediately. The dashboard screenshots below show the system state under each of the three evaluation scenarios described in Section 5.6.

4.17 End-to-End Pipeline Operation

4.17.1 Integrated Pipeline Operation

Each monitored smart-home device generates system-level and network-level measurements, referred to as features, which are collected as paired data at each sampling interval. These measurements are combined into the integrated hybrid feature vector defined earlier in this chapter, which is subsequently assessed against the device-specific behavioural baseline established during the profiling stage. Deviation scores are then produced in relation to the normal behaviour of the specific device being monitored. This feature-level evidence is passed to the profiling-grounded anomaly-detection component and provides the statistical context required by the subsequent detection and assurance components.

The anomaly-detection stage checks the current observation to determine whether its behaviour is consistent with the normal operating profile or requires further analysis for security-posture assessment, as represented by Equation (4.25):

$$\text{Behaviour_State}(t) = \text{RF}_{\text{AD}}\big(X(t)\big). \tag{4.25}$$

Observations that are not considered anomalous continue to contribute to continuous behavioural monitoring and security-posture calculation through the other components of the implemented pipeline. The anomaly-detection function is not part of STRIDE itself. When an observation requires escalation, the intrusion-detection classifier assigns a supported attack category, as represented by Equation (4.26):

$$\text{Attack_Type}(t) = \text{RF}_{\text{IDS}}\big(X(t)\big). \tag{4.26}$$

The verification layer combines three physical-consistency rules with two Isolation Forest confirmation gates. The temporal rule assesses deviations from established statistical behaviour, while the causal rule checks whether the observation is consistent with relationships between relevant system and network measurements. The STRIDE-semantic rule evaluates whether the classified attack is feasible according to the available feature representation. The Isolation Forest gates provide complementary checks against their reference distributions. This verification phase is intended to detect inconsistencies that may not be identified by the classifier-only decision because of adversarial manipulation of the features.

The monitoring process concludes with normalisation of deviations, weighted accumulation of security-relevant evidence within the security-posture quantification model, and transformation of that evidence into the bounded CSA score, as described in Equations (4.22)–(4.24). The updated security posture is subsequently represented through the monitoring dashboard, allowing the operational security condition to be assessed continuously. Chapter 5 presents the quantitative evaluation of the effectiveness of these mechanisms.

Chapter Summary

The chapter has introduced how the proposed framework combines synchronised Sys-Net telemetry, device-specific behavioural profiling, anomaly detection, STRIDE-aligned intrusion classification, physical consistency verification, and continuous security posture quantification. In terms of design, the assessment of behavioural deviation is separated from semantic threat interpretation, and the behavioural risk information is combined within a common monitoring framework. SmartHome-StrideDS provides the primary experimental foundation for the detection and operational security components, while the verification layer is intended to assess the integrity of the detection and posture information. After describing the implementation and end-to-end processing sequence, the framework is evaluated through controlled baseline experiments, modality ablation, scenario-based evaluation, adversarial testing, and cross-dataset experiments.

Chapter 5

Experimental Results and Validation

5.1 Introduction

This chapter reports an empirical evaluation of the proposed Continuous Security Assurance (CSA) framework. Unlike conventional approaches that rely exclusively on either host-level or network-level metrics, the CSA model is designed to exploit correlated, cross-layer measurements in order to detect elusive indicators of compromise that may be distributed across multiple system components. This multi-dimensional perspective is of particular importance in the modern IoT threat landscape, where adversaries routinely employ concealment strategies such as living-off-the-land (LotL) execution or low-and-slow exfiltration to circumvent conventional detection protocols (Pang et al., 2021; Lo et al., 2026).

The evaluation methodology is twofold. First, a statistical characterisation of the SmartHome-StrideDS dataset is presented, documenting how host-level and network-level characteristics behave under both normal conditions and active attacks. Second, a rigorous performance comparison of several machine-learning classifiers is conducted. A central contribution of this experimental work is quantifying the practical benefits of multimodal data integration by conducting comparative experiments on network-only, host-only, and fully synchronised hybrid datasets.

5.2 Dataset Characteristics and Profiling Validation

Table 5.1: Feature Statistics (Mean ± Std. Dev.) by Class

Feature	**Normal**	**DoS**	**Spoofing**	**Tampering**	**Info. Disc.**
cpu (%)	12.7 ± 11.7	66.2 ± 41.9	22.4 ± 20.7	22.7 ± 20.8	21.6 ± 21.3
memory (%)	29.2 ± 17.2	35.4 ± 15.4	35.7 ± 15.5	35.7 ± 15.5	35.4 ± 15.4
disk	79.3 ± 12.0	81.9 ± 12.2	83.3 ± 12.2	83.3 ± 12.2	83.9 ± 12.2
load (1-min avg)	0.4 ± 0.3	6.2 ± 6.9	0.8 ± 0.5	0.7 ± 0.5	3.9 ± 4.8
uptime (10^5 s)	2.0 ± 1.1	1.2 ± 0.9	1.3 ± 0.9	1.3 ± 0.9	1.3 ± 0.9
bytes sent (10^8)	6.6 ± 4.2	7.5 ± 9.6	8.8 ± 10.6	8.8 ± 10.6	9.4 ± 10.9
bytes recv (10^8)	10.6 ± 10.5	5.5 ± 4.7	5.8 ± 4.8	5.8 ± 4.8	6.2 ± 4.8
tcp conn	6.1 ± 2.9	5.8 ± 2.3	5.9 ± 2.3	5.9 ± 2.5	5.8 ± 2.2
icmp packets	59487 ± 35000	32230 ± 24167	33019 ± 24029	33051 ± 23932	32650 ± 23285
open ports	161 ± 263	1891 ± 3158	2372 ± 3446	2383 ± 3446	2601 ± 3538
latency (ms)	19.4 ± 25.6	21.9 ± 25.6	21.2 ± 25.2	21.5 ± 25.1	21.1 ± 23.2

5.2.1 Validation via Baseline Anomaly Detection

To isolate the contribution of device-specific profiling from the downstream classifier, a direct comparison was conducted on SmartHome-StrideDS using the statistical profiles alone, without any supervised learning. Two detection strategies were evaluated. The conventional global threshold approach computed $\mu + 2\sigma$ boundaries across the pooled dataset, treating all nine devices as a single population, the standard practice in deployments that do not distinguish between device roles. The proposed per-device approach estimated separate (μ_j, σ_j) pairs for each node from its own benign observations. The results are reported in Table 5.2.

Table 5.2: Traditional Global Threshold vs. Proposed Per-Device Hybrid Profiling

Metric	**Traditional Baseline**	**Proposed Profiling**	**Improvement**
Accuracy (%)	78.4	94.7	+16.3
F1-score (%)	76.2	93.9	+17.7
Precision (%)	81.5	95.8	+14.3
Recall (%)	71.8	92.1	+20.3
False Positive Rate (%)	18.6	3.9	−14.7

The magnitude of the gap across every metric reflects a structural problem with the pooled-threshold approach rather than an incidental shortcoming of any particu-

lar parameterisation. When nine devices with genuinely different workload profiles are pooled, the resulting distribution is heterogeneous in ways that no single threshold can accommodate: a boundary calibrated conservatively enough to contain the high-activity nodes will be far too liberal for the low-activity ones, and vice versa. In this dataset, that mismatch manifests as an 18.6 % false positive rate for the global approach. An analyst who encounters a false positive in roughly one out of every five alerts will rapidly adjust their response behaviour accordingly, and genuine intrusions will be discounted alongside the spurious ones. Reducing the false positive rate to 3.9 % through device-specific profiling is therefore not merely a metric improvement but a qualitative shift in the operational usability of the detection output. The supervised machine-learning evaluation, in which Random Forest is applied to the per-device profiles across the full SmartHome-StrideDS dataset, is reported in this chapter, where the combined framework achieves 96.42 % accuracy.

5.3 Anomaly Detection Results

5.3.1 Hybrid Sys-Net Dataset

The evaluation focuses on DoS and Spoofing, the two attack classes whose signatures span both telemetry layers most distinctly. Denial-of-service conditions produce sustained elevation of CPU and memory utilisation; spoofing attacks distort network communication metadata and connection patterns while leaving system-level resources essentially undisturbed. Per-attack recall on these two classes provides a targeted test of whether the hybrid feature space provides discriminative power that neither plane would yield alone.

kNN and LOF achieve recall figures well above 70 % for both attack classes, indicating that the joint system–network feature space places attack samples at a measurable distance from the benign distribution. HBOS and Isolation Forest fall notably short, particularly on Spoofing, where both methods produce recall figures barely above chance.

The HBOS shortfall is consistent with its independence assumption, which prevents detection of the coordinated multi-feature shifts that characterise spoofing traffic. The weaker Isolation Forest result is less immediately expected given its general robustness to high-dimensional data, but is consistent with prior reports that ensemble isolation methods can miss attacks whose feature-space footprint overlaps substantially with dense benign regions (Liu, Ting, and Zhou, 2008).

5.3.2 TON Linux Process Dataset

The TON Linux Process data originates from the TON-IoT Linux dataset family described by Moustafa, M. Ahmed, and S. Ahmed (2020); the process-level evaluation reported here was generated independently in this thesis and does not reproduce the published paper's Linux disk-data anomaly experiment.

Repeating the evaluation under the restricted single-plane condition provided by the TON Linux process dataset produces the results in Table 5.7.

The degradation across all four methods is severe and consistent. kNN DoS recall falls from 89.85 % to 11.48 %; Isolation Forest drops from 72.97 % to 7.15 %. Reductions of this magnitude are not marginal artefacts of parameter sensitivity , they indicate that single-plane process-level telemetry is structurally insufficient to support reliable anomaly detection across the attack categories examined here. The result directly quantifies what the hybrid design of SmartHome-StrideDS was intended to address.

5.4 IDS Classification Results

5.4.1 Comparative Performance of Classifiers

Table 5.3 reports the overall and per-class F1-scores of the candidate classifiers evaluated on the synchronised hybrid SmartHome-StrideDS configuration, together with classification accuracy.

Table 5.3: Per-class F1-score and accuracy comparison of candidate classifiers on SmartHome-StrideDS (Synchronised Hybrid Sys-Net configuration).

Algorithm	Overall F1	Normal	DoS	Spoofing	Tampering	Info. Disclosure	Accuracy
Logistic Regression	0.73	0.93	0.65	0.37	0.36	0.66	74.80%
Random Forest	0.96	1.00	0.96	0.89	0.89	0.97	96.42%
SVM (RBF Kernel)	0.77	0.95	0.68	0.37	0.50	0.71	77.00%
MLP	0.78	0.96	0.68	0.38	0.51	0.71	79.00%

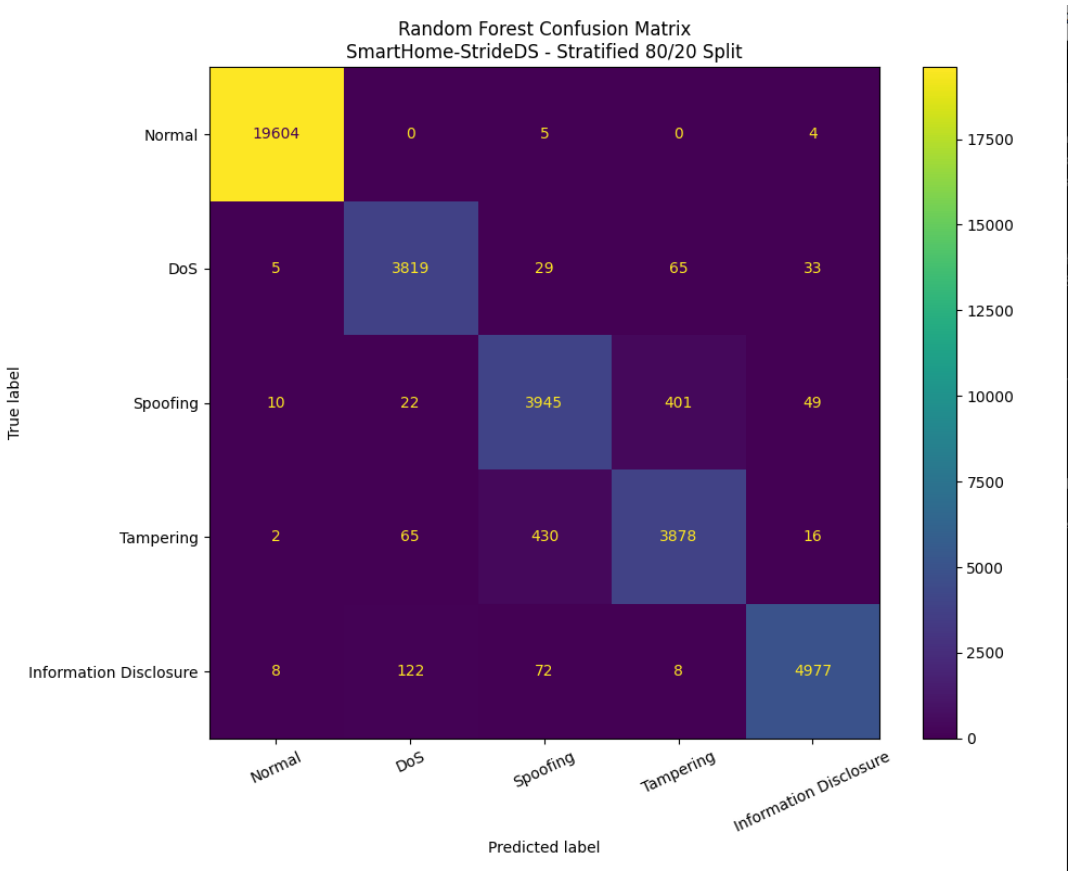

Figure 5.1: Overall, Random Forest was the best performer with an overall F1-score of 0.96 and accuracy of 96.42%. The average F1-scores were Normal (1.00), DoS (0.96), Spoofing (0.89), Tampering (0.89), and Information Disclosure (0.97) for each class.

The confusion matrix validates good separation of the classes for all five considered classes. Normal and Information Disclosure are identified with particularly high consistencies and the biggest residual confusion can be seen between the two hybrid behavioural patterns Spoofing and Tampering, indicating the similarity between them.

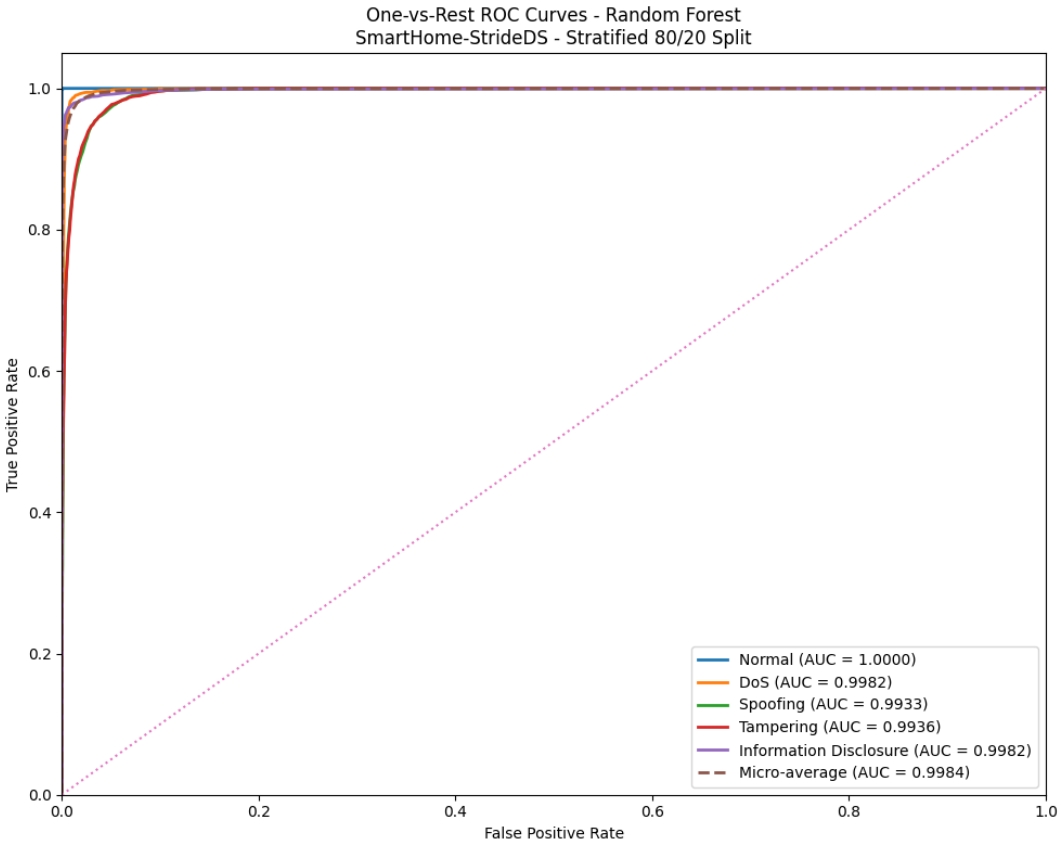

Figure 5.2: The one-vs-rest implementation of the Random Forest (RF) classifier for SmartHome-StrideDS has been plotted based on an 80/20 training-testing split. The AUC for the classifier was micro-average 0.9984 and 0.9933 - 1.0000 for each class respectively.

The results of ROC analysis demonstraes strong discrimination power for all five classes. This minor ambiguity in the classification of certain examples is reflected by lower AUC values, close to 1, for Spoofing (0.9933) and Tampering (0.9936) compared to their counterparts Normal, DoS and Information Disclosure, for which the AUC values were almost 1.0.

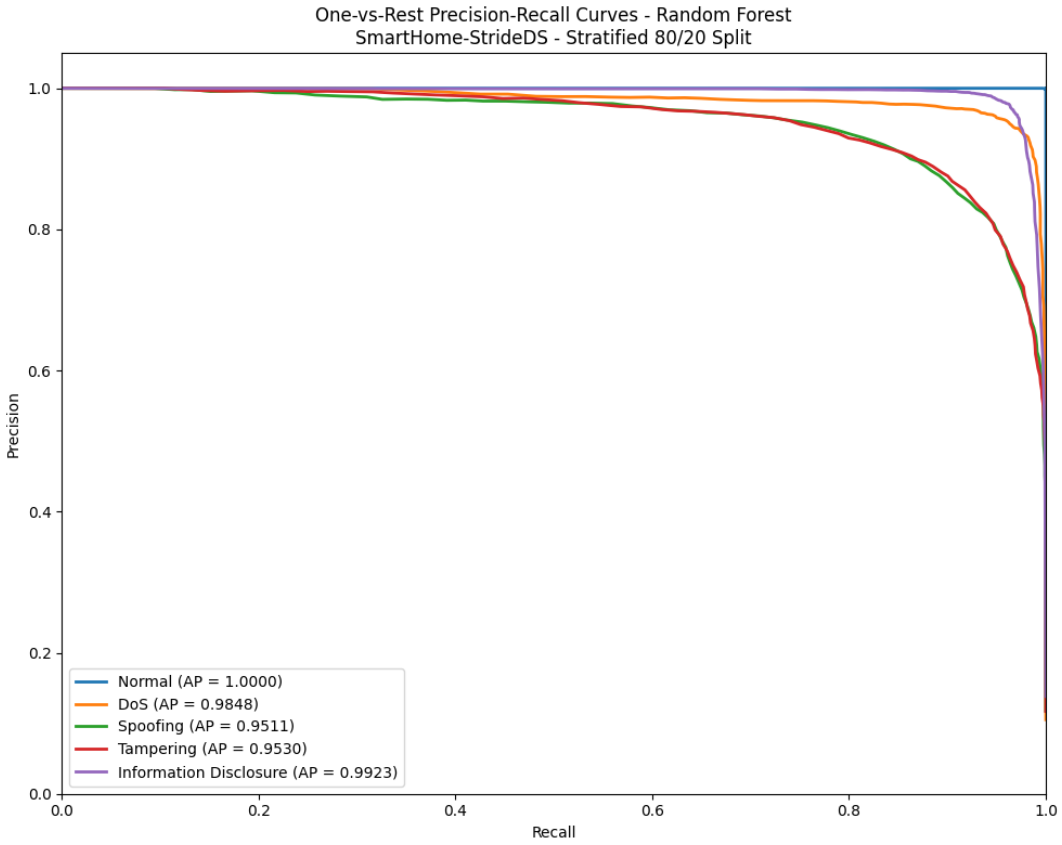

Figure 5.3: One-vs-rest precision-recall curves for the Random Forest classifier when 80/20 train-test split is used for SmartHome-StrideDS. Overall, the classifier had an average-precision of 1.0000 for Normal, 0.9848 for DoS, 0.9511 for Spoofing, 0.9530 for Tampering and 0.9923 for Information Disclosure.

From precision-recall analysis, it is observed that Random Forest Classifier adapts to high precision at most levels of recall for all the five classes. As seen in the residual overlap in the confusion matrix and the ROC, the strongest curves are seen for Normal and Information Disclosure, with slightly lower average-precision seen for Spoofing and Tampering.

Random Forest achieved the strongest overall performance, with an overall F1-score of 0.96 and an accuracy of 96.42%. Its per-class F1-scores were 1.00 for Normal, 0.96 for DoS, 0.89 for Spoofing, 0.89 for Tampering, and 0.97 for Information Disclosure.

Three further observations warrant discussion. Random Forest achieved substantially stronger DoS detection than the other evaluated classifiers, with an F1-score of

0.96 compared with 0.65 for Logistic Regression and 0.68 for both SVM and MLP. The contrast with Spoofing and Tampering is instructive. Both categories generate cross-layer signals that are subtler and more easily confounded with legitimate operational events. Finally, the high F1-score for Information Disclosure merits note: the characteristic pre-exfiltration indicators, elevated outbound byte counts, connections to unfamiliar endpoints, are subtle by design, yet the profiling-guided training input allowed the classifier to detect them reliably.

5.5 Existing Comparison Evidence

The studies listed in the Table 5.4 are not numerically equivalent as they draw on different data sets and attack distributions, as well as different data modalities and data validation methods. These are compared and contrasted, therefore, with recent smart-home IDS, anomaly detection, host-network vs. adversarially robust approaches. TON-IoT studies have been given a greater weight due to the fact that TON-IoT Host-Process and Network-Flow datasets have also been used for independent evaluation in this thesis.

Table 5.4: Comparison of relevant supervised, unsupervised/anomaly-detection and hybrid host–network approaches with the proposed framework; differing datasets and evaluation procedures mean that this is not a direct leaderboard.

Study	Dataset	Data Modality	Model	Attack Classes	Validation Method	Accuracy	Precision	Recall	F1-score	FPR
Moustafa, M. Ahmed, and S. Ahmed (2020)	TON-IoT Linux	Linux host telemetry; published anomaly experiment on disk data	kNN; LOF; HBOS (among other unsupervised algorithms)	DDoS; DoS; Injection; Scanning; Password; XSS; MITM	Unsupervised evaluation on Linux disk data; attack-specific hit rate/TPR reported	NR	NR	NR	NR	NR
Moustafa (2021)	TON-IoT Network / Network-Flow	Network-flow traffic	GBM (RF, NB and DNN also evaluated)	Normal; Backdoor; DDoS; DoS; Injection; MITM; Password; Ransomware; Scanning; XSS	Training/testing subset; binary Normal/Attack; 10-fold CV; IP addresses and ports retained	NR	> 0.99 (GBM)	> 0.99 (GBM)	> 0.99 (GBM)	NR
Meidan et al. (2023)	CADeSH smart-home experimental data	Network/IoT behavioural traffic (IPFIX)	Autoencoder + k-means clustering; collaborative anomaly detection	Nmap scanning; cryptomining execution	Models trained on devices outside the lab; labelled laboratory traffic used as the test set; grid search	NR	0.875	0.992	0.929	0.014
Lara et al. (2023)	AMPds smart-home data + in-house attack data	Network traffic + application-layer data	Network IoCs + application-layer anomaly detectors; Snort complement	Eight network attacks; four False Command/Data Injection attacks	Smart-home case study using public application data and implemented attack tests; Snort detected 4/12 attacks	NR	NR	NR	NR	$< 0.3\%$
Alars and Kurnaz (2024)	Network Intrusion Detection dataset	Combined host + network features	CNN-LSTM deep-learning NIDS	NR	70/15/15 training/validation/test split; 10-fold cross-validation	98.5%	97.8%	96.0%	97.3%	NR
This thesis: binary hybrid Sys-Net anomaly detector	SmartHome-StrideDS	Synchronised Hybrid Sys-Net / Host + Network	Random Forest binary anomaly detector	Normal vs Attack	Stratified 80/20 split; training-set standardisation; Class 0 / Class 1 = 0.9999 / 0.9974	0.9992	0.9992	0.9992	0.9992	NR
This thesis: supervised STRIDE classifier (Table 5.3 setting)	SmartHome-StrideDS	Synchronised Hybrid Sys-Net / Host + Network	Random Forest	Normal; DoS; Spoofing; Tampering; Information Disclosure	Stratified 80/20 split	96.42%	NR	NR	0.96	NR

Notes: NR = Not Reported. External results are reproduced only where the cited publication explicitly reports the requested metric.

In Table 5.4, the numerical results of studies related to the problem are reported; however, differences in datasets, attack categories, feature representations, and evaluation protocols do not allow for a direct ranking of the performance of the various methods. To obtain quantitative evidence in a controlled manner, representative machine-learning baselines were tested on SmartHome-StrideDS under the same experimental configuration (Table 5.3). The Random Forest model performed best with an accuracy of 96.42%, followed by the MLP model with an accuracy of 79.00%, the SVM model with an accuracy of 77.00%, and the Logistic Regression model with an accuracy of 74.80%. This represents an improvement of 17.42 percentage points over the best alternative classifier. The controlled results support the effectiveness of the selected classification approach, while the modality ablation results in Table 5.5 separately assess the value of the selected hybrid Sys-Net telemetry.

Moreover, representative baseline models were evaluated in SmartHome-StrideDS with the same preprocessing and train-test setting, along with the literature comparison. The modality ablations for this reason are a controlled ablation, where the dataset and split is held constant while the type of modality (model) is changed. The classifier comparison and modality ablation therefore provide controlled comparisons in which the dataset and split remain fixed while the model or telemetry modality is varied.

Table 5.3 contains the supervised classifier comparison. Table 5.5 contains the Hybrid / Network-only / Host-only modality comparison. Table 5.7 contains the kNN / LOF / HBOS / Isolation Forest comparison.

5.6 Scenario-Based CSA Evaluation

Three operationally distinct test conditions were constructed, each targeting a different dimension of the framework's capability. The first establishes a baseline by running the system under clean conditions; the second introduces a realistic coordinated attack to

test the primary detection and scoring response; the third presents the most demanding challenge, a black-box adversarial attack mounted by an adversary who has optimised explicitly against the consistency rules.

5.6.1 Normal Operation

Under the baseline condition, all nine devices were left to operate normally throughout a sustained observation period with no injected attack traffic. The per-feature z-scores $z_j(t)$ fluctuated only slightly about zero, reflecting the background variability intrinsic to live hardware and software rather than any genuine anomalous activity. The accumulated risk was correspondingly negligible,

$$\mathrm{Risk}(t) \approx 0, \tag{5.1}$$

which by Equation (4.24) placed the posture score in the vicinity of its nominal ceiling,

$$\mathrm{CSA}(t) \approx 100. \tag{5.2}$$

In practice, per-device scores settled between 95 % and 99 % rather than reaching exactly 100; this offset is expected and appropriate, since the model is not designed to suppress the small but genuine variability that real hardware produces. The z-scores across all features throughout this scenario clustered between 0.03 and 0.07. T_{EWMA} logged no violations over the entire observation window. C_{corr} found no broken causal pairs. S_{STRIDE} encountered no suppressed signature features, for the simple reason that no attack was present to generate them. With $\mathrm{Risk}(t) \approx 0.041$ the aggregate score settled at approximately 96 %, confirming that the model neither penalises ordinary operational variability nor requires a perfectly noise-free environment to behave sensibly.

5.6.2 Standard STRIDE Attack

The second scenario introduced a coordinated, multi-category intrusion. Four devices were targeted simultaneously: Denial of Service was directed at IoT-02, Spoofing at IoT-04, Tampering at IoT-06, and Information Disclosure at IoT-08. Each attack ran for ten minutes without interruption.

The framework's response was immediate and unambiguous. All four targeted devices crossed multiple anomaly thresholds within the first sampling interval following attack onset. T_{EWMA} flagged all four immediately: the attack-induced changes were large enough to displace considerably more than half the monitored features beyond three standard deviations of their respective class means. S_{STRIDE} confirmed in parallel that the signature features characteristic of each attack type remained well above the 50 % viability threshold, indicating that the attacks were operationally genuine rather than engineered to minimise their visible footprint. C_{corr} additionally recorded broken causal pairs for the Spoofing and Denial of Service events, in both of which the attacker's traffic had disrupted the expected co-movement between connection count and latency. The master decision gate blocked all four devices' attack samples in the first decision epoch.

The aggregate posture score fell from 96 % to 43 % at attack onset and recovered to 91 % once injected traffic ceased. The recovery is as significant as the fall: the score is not inertially sticky, and it does not maintain a degraded value after the threat has resolved. A system that held its score below nominal after an attack ended would progressively lose its value as an indicator of the current state of the network, which would represent a distinct kind of operational failure even if the detection capability remained intact.

5.6.3 Adaptive Adversarial Attack

The third scenario is the one from which the most important conclusions about the framework's robustness can be drawn. A coordinated adaptive black-box attack was

directed simultaneously at three devices by an adversary who had explicitly encoded each of the three consistency rules as a penalty term within the perturbation objective. The adversary was not merely trying to fool the classifier; it was trying to fool the classifier while simultaneously remaining within the bounds that Rules 1, 2, and 3 enforce.

In its primary objective the attacker succeeded: T_{EWMA}, C_{corr}, and S_{STRIDE} all remained silent throughout the attack window. Not one of the three physical verification rules triggered. The attacks were, however, detected, and this is the result on which the framework's claim to robustness rests.

The reason is geometric. To simultaneously satisfy the constraints imposed by Rules 1 to 3 and flip the classifier's label, the adaptive attacker was forced to move the perturbed samples into a region of feature space that is improbable by the standards of both known distributions: the low-density void lying between the attack-class manifold and the normal-traffic manifold. Both Isolation Forest gates registered the displacement. IF_{src} returned -1 because the perturbed vectors no longer resembled any known member of their claimed attack class; IF_{norm} returned -1 because those same vectors were equally dissimilar to clean normal traffic. This satisfied Condition 3 of Equation (4.20), which, critically, imposes no requirement that any physical check have fired. The aggregate posture score settled at 61 %, sufficient to trigger a high-priority operator alert. The dimensional dead-space mechanism is architecturally independent of the rule-based layer; it identifies adversarial inputs through the geometry of feature space alone.

Figures 5.4–5.6 show the existing dashboard observations for normal operation, a coordinated STRIDE attack, and an adaptive adversarial attack.



Figure 5.4: CSA dashboard during normal operation.

5.7 Validating the Sys-Net Approach: A Quantitative Comparison of Data Modalities

To test the central hypothesis quantitatively, that synchronised Sys-Net data yields superior detection performance, a comparative ablation study was conducted. A Random Forest was trained and tested on four distinct dataset configurations:

1. **Synchronized Hybrid (SmartHome-StrideDS):** The multimodal dataset incorporating all host-level and network-level features.
2. **Network-only (Subset):** A version from which all host-level features were removed.
3. **Host-only (Subset):** A version from which all network-level features were removed.
4. **Network-only & Host-only (Illustrative – ToN_IoT):** A demonstrative baseline reflecting typical real-world single-modality deployment.



Figure 5.5: CSA dashboard during a coordinated STRIDE attack.

Three Random Forest models were trained independently using identical hyperparameters (500 estimators, Gini criterion, stratified 80/20 split) on three distinct feature subsets derived from SmartHome-StrideDS. In all three cases the same stratified 80/20 split was applied to the same labelled rows, ensuring that performance differences are attributable solely to feature-plane availability.

Table 5.5: Quantitative Comparison of Modalities Against STRIDE Attacks (Random Forest F1-Scores)

Dataset Configuration	**Data Modality**	**DoS**	**Spoofing**	**Tampering**	**Info. Disc.**
SmartHome-StrideDS	Synchronized Hybrid	0.96	0.89	0.89	0.97
SmartHome-StrideDS (subset)	Network-only	0.83	0.86	0.84	0.92
SmartHome-StrideDS (subset)	Host-only	0.86	0.69	0.71	0.86

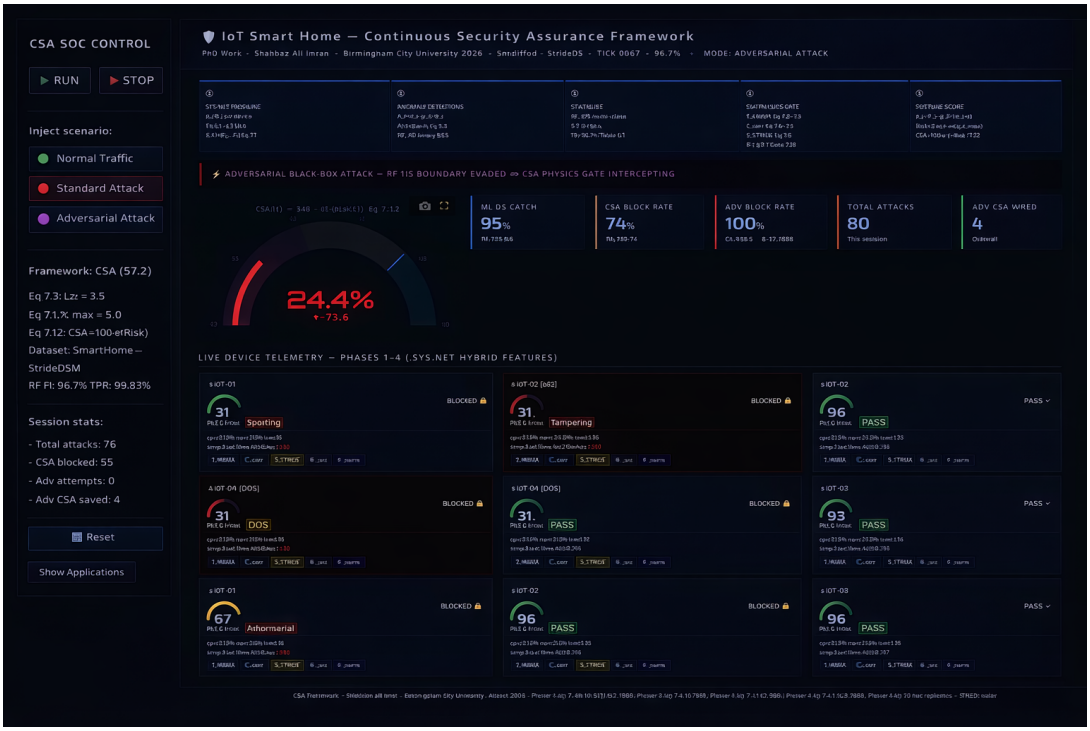

Figure 5.6: CSA dashboard during an adaptive adversarial attack.

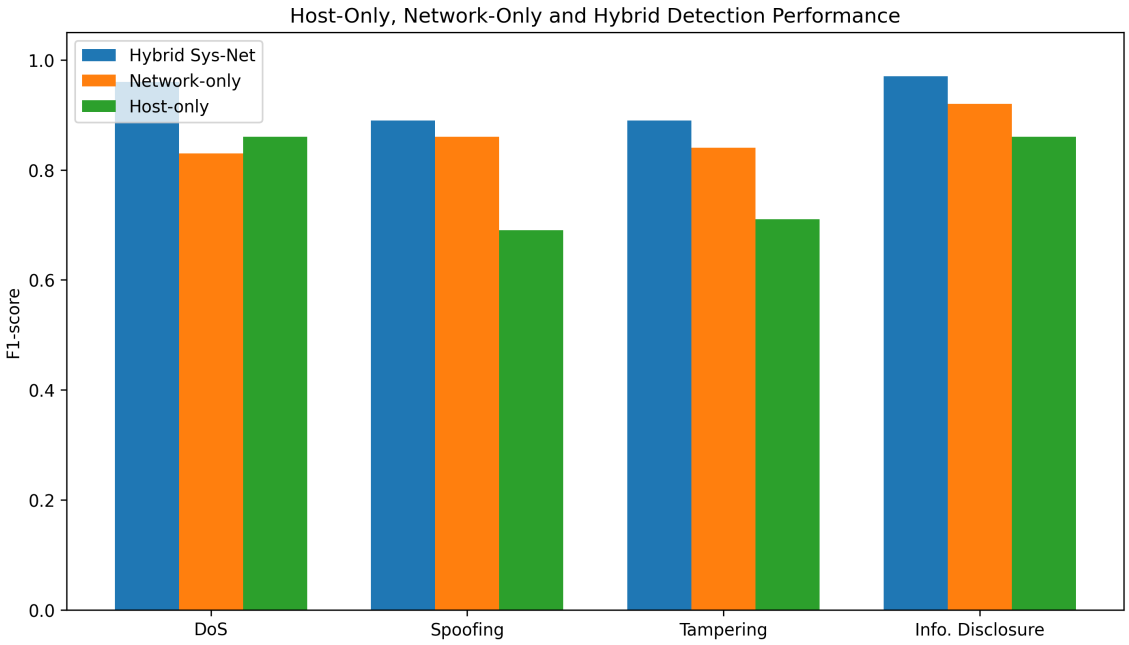

Figure 5.7: RF F1-scores for the synchronised Hybrid Sys-Net Net-Host and Host-Only configurations on SmartHome-StrideDS per class.

The results of the modality comparison reveal the Hybrid- Sys-Net configuration with synchronisation provides consistently high F1 scores over all the analyzed attack types. Spoofing/Tampering shows the biggest change when using just a single telemetry

plane, indicating the need to employ both host and network level evidence.

The results admit an unambiguous interpretation. The model trained on the synchronised hybrid features achieves the highest and most consistent F1-scores across all four attack classes. This outcome is consistent with the broader IDS literature demonstrating that hybrid, multimodal systems possess greater strength and breadth against diverse threats than their single-modality counterparts (Pang et al., 2021).

5.8 Classifier Results and Model Selection

Empirical findings demonstrate that the Random Forest classifier achieves the highest overall performance, with an overall F1-Score of 0.96 and markedly superior detection of the more subtle attack types, specifically Spoofing (89.0%) and Tampering (89.0%), compared with all other evaluated models. The relatively lower performance of Logistic Regression is expected, as it assumes a linear feature-outcome relationship. Random Forest, as an ensemble of many decision trees, effectively addresses the overfitting susceptibility of single trees and captures complex non-linear feature interactions (Breiman, 2001).

5.8.1 Comparative Evaluation: Baseline Methods vs. Proposed Framework

Observed Anomaly Detection Performance (Binary Setting)

Table 5.6 summarises the anomaly detection performance of the Random Forest classifier under a binary setting (Normal vs. Attack) for the hybrid Sys-Net dataset.

Table 5.6: Binary Anomaly Detection Performance (Hybrid Sys-Net Dataset)

Metric	**Test Accuracy**	**Precision**	**Recall**	**F1-score**	**Class 0 / Class 1**
Value	0.9992	0.9992	0.9992	0.9992	0.9999 / 0.9974

Per-Attack Recall Comparison

Table 5.7 presents per-attack recall for unsupervised models on the hybrid Sys-Net dataset and TON Linux dataset.

Table 5.7: Per-Attack Recall (%) — Hybrid Sys-Net vs. TON Linux Dataset

	Hybrid Sys-Net		**TON Linux (Process)**	
Model	**DoS**	**Spoofing**	**DoS**	**MITM**
kNN	89.85	86.53	11.48	38.39
LOF	87.34	87.86	15.55	43.75
HBOS	65.29	54.96	3.60	14.29
Isolation Forest	72.97	54.50	7.15	7.14

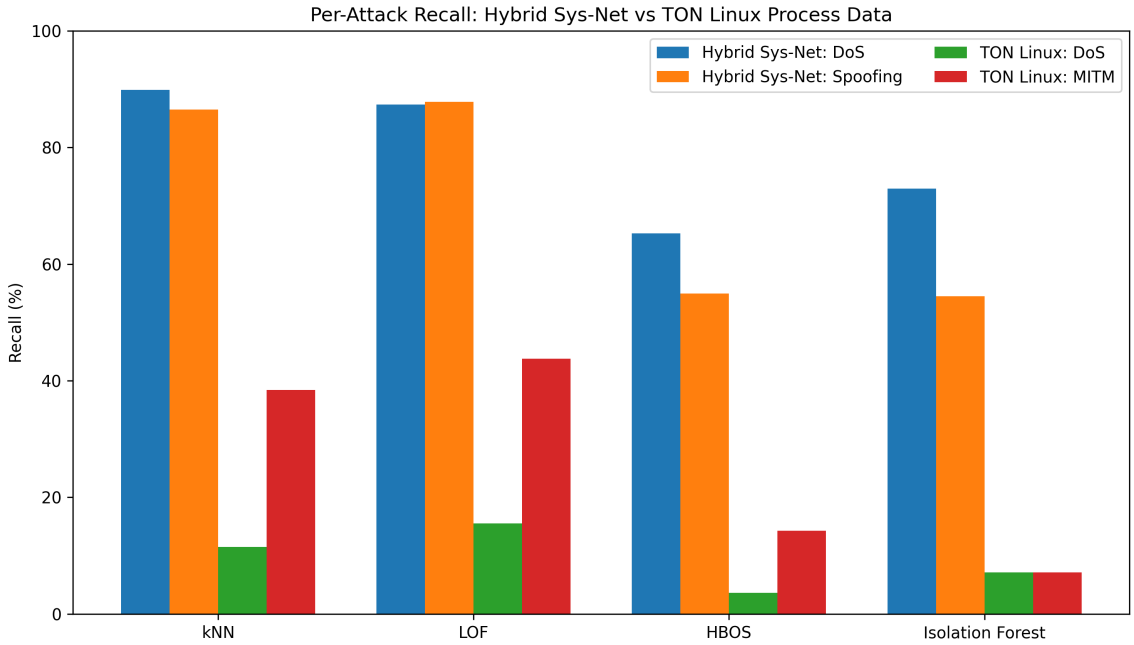

Figure 5.8: Recall (Unsupervised) Per-Attack of hybrid SmartHome-StrideDS configuration and TON-IoT Linux process configuration.

The figure demonstrates the sizeable gap between the case for the hybrid SmartHome-StrideDS setup and for the limited case study: TON-IoT - "process-only". The most promising results are observed for DoS and Spoofing for the hybrid data for kNN and LOF, while recall drops significantly for the TON-IoT process data, especially for DoS.

Cross-Dataset Binary Anomaly-Detection Performance Comparison

Table 5.8 provides a cross-dataset comparison of binary anomaly-detection performance between the network-only TON-IoT condition and the proposed Hybrid Sys-Net CSA condition. This comparison concerns Normal-versus-Attack detection and is separate from the multiclass STRIDE classification experiment reported in Table 5.3.

Table 5.8: Cross-dataset binary anomaly-detection performance comparison between TON-IoT network-only and Hybrid Sys-Net CSA configurations.

Dataset / Framework	Overall Accuracy	Macro F1	Precision	Recall
ToN-IoT (Network-Only)	0.9934	0.94	—	MITM: 0.81 / 0.84
Hybrid Sys-Net CSA	0.9958	0.99	≥ 0.99	≥ 0.99

Table 5.8 concerns binary Normal-versus-Attack anomaly detection, not multiclass STRIDE classification. Random Forest is used in both compared configurations: TON-IoT represents the network-only comparison condition, whereas Hybrid Sys-Net combines host-level and network-level evidence. The MITM precision and recall values of 0.81 and 0.84 are attack-specific diagnostic values reported for the TON-IoT condition. Because the configurations are evaluated across different datasets, the comparison is contextual and should not be interpreted as a controlled same-dataset leaderboard. Table 5.3 remains the authoritative multiclass STRIDE IDS result. The Hybrid Sys-Net value reported in Table 5.8 belongs to this cross-dataset comparison and should not be conflated with the separately reported SmartHome-StrideDS binary result in Table 5.6 or the multiclass STRIDE result in Table 5.3.

5.8.2 Fragility of Purely Statistical Machine Learning Under Adversarial Pressure

The decision-based black-box perturbation experiments expose a structural vulnerability in unconstrained statistical classifiers. Across every evaluated STRIDE class, the gradient-free optimiser located perturbations within the per-feature L^{∞} epsilon budget that repositioned each sample across the Random Forest decision boundary, producing

a confident Normal prediction for what remained, in physical terms, an active attack. Because the Random Forest has no representation of what a physically coherent observation looks like, it cannot distinguish a statistically repositioned attack vector from genuine normal traffic.

This outcome instantiates a fundamental adversarial paradox: the perturbation budget necessary to deceive the statistical classifier is, by construction, larger than the budget that preserves the physical integrity of the attack. The attacker is forced into a choice between a functioning attack that the classifier detects and a statistically invisible attack that cannot execute.

5.8.3 Preservation of Network Causality Under Feature Manipulation

A secondary evasion pathway exploits the conditional independence assumptions of the Random Forest. The causal correlation gate (C_{corr}, Section 4.14.2) is designed precisely to close this pathway. In the Spoofing class, four of the five evaluated examples triggered C_{corr} because the optimiser inflated latency to values exceeding 900 ms whilst simultaneously reducing `tcp_conn` below 3.0. The directional violation (latency rising, `tcp_conn` falling) constitutes a breach of the physical causality rule encoded in Equation (4.17) and is flagged regardless of whether either individual feature value, considered in isolation, falls within a plausible range.

5.8.4 Dimensional Dead Space and the Isolation Forest Confirmation Gates

The Isolation Forest confirmation gates (IF_{src} and IF_{norm}, Section 4.14.4) fired on virtually every successful ML evasion across all five attack classes. The adversarial samples land in a region anomalous with respect to both the source attack class (flagged by IF_{src}) and

genuine normal traffic (flagged by IF_{norm}). This region constitutes the dimensional dead space. When both fire simultaneously (Pathway III of the master gate, Equation (4.20)), detection proceeds without any physical consistency signal, providing a complementary safety net operating independently of the rule-based layer.

5.9 Adversarial Robustness Evaluation: Forensic Breakdown

To demonstrate the operational efficacy of the CSA physical consistency verification layer, Table 5.9 presents a forensic breakdown of twenty-five representative adversarial evasion attempts, five drawn from each STRIDE attack class and from the Normal class, evaluated under the oblivious attacker model at scaling factor $\varepsilon = 0.15$. In every case the primary Random Forest classifier was successfully deceived; the table records the perturbations applied, the CSA gates that subsequently fired, and the physical rationale for detection.

Table 5.9: Forensic breakdown of adversarial evasion and CSA physical detection across all twenty-five evaluated examples (five per class). Gates: `stride` = S_{STRIDE}; `corr` = C_{corr}; `src_if` = source-class IF; `norm_if` = normal-class IF.

True class	**ML pred.**	**Key perturbations**	**Gates fired**	**Detection rationale**
Tampering (Ex. 1)	Normal	load: 3.78 → 0.00; latency: 25.73 → 906.04	stride, src_if, norm_if	load suppressed to zero removes all process activity required for host-level tampering execution. Semantic gate fires; both IFs confirm dimensional anomaly.

Continued on next page

True class	**ML pred.**	**Key perturbations**	**Gates fired**	**Detection rationale**
Tampering (Ex. 2)	Normal	icmp_packets: 535 → 20,178; latency: 160.65 → 637.65	stride, src_if, norm_if	Despite inflated packet counts, load and cpu remain below the 50% semantic threshold. Joint feature shape is anomalous to both class distributions.
Tampering (Ex. 3)	Normal	cpu: 4.40 → 19.40; latency: 36.80 → 873.89	stride, src_if, norm_if	load remains below threshold despite cpu inflation. Extreme latency inflation with no supporting congestion indicators places vector in dimensional dead space.
Tampering (Ex. 4)	Normal	icmp_packets: 1,393 → 2; latency: 14.11 → 941.49	corr, stride, src_if, norm_if	Latency rises whilst tcp_conn drops and icmp_packets collapses. All three causal rules violated simultaneously. Semantic gate fires; both IFs confirm anomaly.
Tampering (Ex. 5)	Normal	load: 0.33 → 0.00; latency: 18.73 → 559.42	stride, src_if, norm_if	load driven to zero. Structural tendency of gradient-free evasion to suppress this feature confirmed. Semantic and dimensional gates fire in concert.
Spoofing (Ex. 1)	Normal	tcp_conn: 5.00 → 2.28; latency: 42.88 → 900.10	corr, src_if, norm_if	Latency inflated to 900 ms whilst TCP connections fall. Causal co-movement violated across all encoded rules. Both IFs confirm dimensional anomaly.

Continued on next page

True class	**ML pred.**	**Key perturbations**	**Gates fired**	**Detection rationale**
Spoofing (Ex. 2)	Normal	tcp_conn: 4.00 → 2.16; latency: 24.29 → 951.67	corr, src_if, norm_if	Latency driven to 951 ms; load simultaneously suppressed. Congestion of this magnitude cannot coexist with falling connection counts and zero system load. Three causal rules breached.
Spoofing (Ex. 3)	Normal	cpu: 12.30 → 0.00; latency: 31.13 → 531.77	corr, src_if, norm_if	cpu zeroed whilst latency inflates seventeen-fold. Zero processor utilisation during active network congestion violates IoT operational consistency.
Spoofing (Ex. 4)	Normal	tcp_conn: 9.00 → 2.55; latency: 20.44 → 396.58	src_if, norm_if	No individual physical consistency rule reaches its threshold, yet both IFs flag the vector. Pathway III of the master gate provides detection independently of the rule-based layer.
Spoofing (Ex. 5)	Normal	tcp_conn: 7.00 → 2.00; latency: 14.90 → 942.28	corr, src_if, norm_if	Latency reaches 942 ms with TCP connections at observed minimum. All three causal rules violated; vector is anomalous to both Spoofing and Normal distributions.

Continued on next page

True class	**ML pred.**	**Key perturbations**	**Gates fired**	**Detection rationale**
DoS (Ex. 1)	Normal	icmp_packets: 51,709 → 31,525; cpu: 13.90 → 28.90	stride, norm_if	Packet volume halved to cross classifier threshold; resulting count falls below minimum flood rate required to sustain resource exhaustion. Semantic gate fires.
DoS (Ex. 2)	Normal	tcp_conn: 5.00 → 2.00; latency: 15.25 → 589.02	src_if, norm_if	Packet volume remains high, so semantic gate does not fire. Extreme latency with falling TCP connections produces a feature shape anomalous to both DoS and Normal distributions (Pathway III).
DoS (Ex. 3)	Normal	icmp_packets: 65,924 → 45,740; latency: 30.98 → 958.35	corr, src_if, norm_if	Packet volume reduced by 30%; latency inflated to 958 ms whilst tcp_conn rises. Causal rule violated. All three gates fire.
DoS (Ex. 4)	Normal	load: 0.24 → 4.59; tcp_conn: 8.00 → 5.05	src_if, norm_if	No physical consistency rule individually reaches threshold; both IFs independently flag the combined feature shape as anomalous, confirming the dimensional dead-space mechanism.
DoS (Ex. 5)	Normal	load: 0.42 → 4.77; latency: 14.59 → 216.50	src_if, norm_if	Packet volume increases while memory drops sharply. No single rule triggered, yet joint multivariate shape is caught by both Isolation Forests.

Continued on next page

True class	**ML pred.**	**Key perturbations**	**Gates fired**	**Detection rationale**
Info. Disc. (Ex. 1)	Normal	cpu: 5.60 → 0.00; latency: 30.53 → 602.04	stride, src_if, norm_if	CPU zeroed: a device executing pre-exfiltration reconnaissance requires active processor engagement. Zero utilisation is physically impossible during execution.
Info. Disc. (Ex. 2)	Normal	load: 1.22 → 0.00; latency: 36.78 → 570.75	stride, src_if, norm_if	load and cpu move in opposite directions; load suppressed to zero whilst cpu inflates. Semantic gate fires; both IFs confirm dimensional anomaly.
Info. Disc. (Ex. 3)	Normal	cpu: 5.60 → 0.00; latency: 101.61 → 737.06	corr, stride, src_if, norm_if	All four gates fire simultaneously. CPU zeroed; latency inflates seven-fold; icmp_packets collapses whilst tcp_conn rises. Three causal rules breached.
Info. Disc. (Ex. 4)	Normal	icmp_packets: 9,419 → 2; latency: 13.83 → 0.00	stride, norm_if	Both icmp_packets and latency driven to minimum; cpu inflates but does not reach semantic elevation threshold. Semantic gate fires; norm_if confirms anomaly.
Info. Disc. (Ex. 5)	Normal	icmp_packets: 96,017 → 80,132; latency: 21.41 → 0.00	stride, norm_if	Packet volume reduced; latency suppressed to zero whilst load inflates. Zero network latency on an active IoT device is physically incoherent.

Continued on next page

True class	**ML pred.**	**Key perturbations**	**Gates fired**	**Detection rationale**
Normal (Ex. 1)	Spoofing	icmp_packets: 25,843 → 46,027; latency: 15.43 → 720.01	src_if, norm_if	No physical consistency rule fires since original class is Normal. Both IFs catch dimensional anomaly: vector sits outside Normal and Spoofing distributions simultaneously.
Normal (Ex. 2)	Spoofing	cpu: 10.80 → 0.00; icmp_packets: 23,631 → 43,815	src_if, norm_if	CPU zeroed whilst packet count nearly doubles and load inflates. Feature shape is anomalous to the Normal distribution; src_if confirms it does not resemble normal traffic.
Normal (Ex. 3)	Tampering	icmp_packets: 6,122 → 26,306; latency: 14.71 → 942.09	src_if, norm_if	Normal sample driven toward Tampering geometry. Latency inflated sixty-four-fold; load suppressed to zero. Perturbed vector does not resemble Tampering nor genuine benign traffic.
Normal (Ex. 4)	Tampering	cpu: 4.50 → 0.00; latency: 14.87 → 687.06	corr, src_if, norm_if	Latency driven to 687 ms; tcp_conn drops and icmp_packets falls. Causal rule violation. All three gates fire.

Continued on next page

True class	ML pred.	Key perturbations	Gates fired	Detection rationale
Normal (Ex. 5)	Spoofing	tcp_conn: 7.00 → 2.35; latency: 17.33 → 904.09	corr, src_if, norm_if	Latency at 904 ms with TCP connections at observed minimum and cpu zeroed. Three causal rules breached. Vector is anomalous to both Normal and Spoofing distributions.

5.10 Adversarial Robustness Evaluation

The evaluation reported here examines the degree to which bounded feature-space perturbations can circumvent the proposed two-layer defence, an adversarially trained Random Forest paired with the CSA physical consistency verification layer (Sections 4.14 and 5.9), under controlled experimental conditions. Two metrics are reported throughout. The Machine Learning Attack Success Rate (ML-ASR) measures the proportion of test samples that successfully relocate across the classifier decision boundary within the permitted per-feature ℓ^∞ budget $\varepsilon_f = \alpha \cdot (\max(f) - \min(f))$, $\alpha \in \{0.01, 0.05, 0.10, 0.15\}$ (Equation (3.4)). The CSA-Net Attack Success Rate (CSA-ASR) is the fraction that simultaneously fools the classifier and evades every gate of the consistency verification layer. The difference $\Delta = \text{ML-ASR} - \text{CSA-ASR}$ constitutes the CSA absolute reduction, the primary measure of the physical consistency layer's operational contribution.

Two attacker models are maintained throughout (Biggio and Roli, 2018; Papernot et al., 2017). An oblivious attacker optimises only against the Random Forest with no representation of the CSA gates. An adaptive attacker incorporates explicit penalty terms for gate violations into the attack loss (penalty weights: $w_{\text{EWMA}} = 2.0$, $w_{\text{corr}} = 3.0$, $w_{\text{STRIDE}} = 2.0$), constituting the most informed threat model evaluated short of full white-box gradient access.

5.10.1 Classifier Fragility and the Contribution of Adversarial Training

Table 5.10 presents the full oblivious-attacker results for both classifier variants across all four perturbation budgets. At $\varepsilon = 0.15$ the overall ML-ASR for RF_vanilla reaches 66.6%, confirming that a gradient-free random-search optimiser is sufficient to relocate the majority of attack-class samples across a Random Forest decision boundary (Kantchelian, Tygar, and Joseph, 2016). The CSA consistency verification layer reduces this to a CSA-ASR of 22.2% (2,217 attacks blocked), a 44.3 pp reduction. Adversarial training substantially contracts the initial attack surface: `RF_advtrain` at $\varepsilon = 0.15$ achieves an overall ML-ASR of only 16.5%. The CSA layer then further reduces the residual evasion to 6.4% (505 attacks blocked), giving an overall two-layer residual attack success rate below one-in-sixteen attempts.

Table 5.10: Oblivious attacker: ML-ASR and CSA-ASR (%) for RF_vanilla and RF_advtrain across all perturbation budgets, SmartHome-StrideDS (n = 1,000 per STRIDE class, $N_{\text{iter}} = 1{,}000$). Δ = ML-ASR − CSA-ASR (percentage points).

	$\varepsilon = 0.01$			$\varepsilon = 0.05$			$\varepsilon = 0.10$			$\varepsilon = 0.15$		
STRIDE Class	ML	CSA	Δ	ML	CSA	Δ	ML	CSA	Δ	ML	CSA	Δ
RF_vanilla (standard Random Forest, clean training data)												
Tampering	6.9	6.7	0.2	47.6	41.6	6.0	68.1	42.1	26.0	71.5	29.6	41.9
Spoofing	6.3	6.3	0.0	49.0	45.5	3.5	69.7	49.2	20.5	74.6	30.7	43.9
DoS	6.3	6.3	0.0	37.7	29.1	8.6	67.7	24.4	43.3	66.9	19.4	47.5
Info. Disclosure	1.9	1.9	0.0	14.3	12.6	1.7	34.0	23.5	10.5	39.8	17.0	22.8
Normal	1.3	1.3	0.0	46.3	45.1	1.2	61.5	30.4	31.1	80.1	14.5	65.6
Overall	**4.5**	**4.5**	**0.0**	**39.0**	**34.8**	**4.2**	**60.2**	**33.9**	**26.3**	**66.6**	**22.2**	**44.3**
RF_advtrain (adversarially hardened, $\alpha = 0.15$, 10,000 augmented instances per class)												
Tampering	8.8	8.8	0.0	13.8	12.9	0.9	13.1	9.3	3.8	17.4	6.8	10.6
Spoofing	8.9	8.9	0.0	14.5	14.5	0.0	13.8	11.6	2.2	18.7	9.3	9.4
DoS	7.9	7.9	0.0	13.8	13.1	0.7	10.4	7.3	3.1	15.6	8.9	6.7
Info. Disclosure	3.4	3.4	0.0	6.7	6.7	0.0	4.6	3.4	1.2	5.7	2.9	2.8
Normal	3.5	3.4	0.1	6.3	5.9	0.4	8.2	3.7	4.5	25.2	4.2	21.0
Overall	**6.5**	**6.5**	**0.0**	**11.0**	**10.6**	**0.4**	**10.0**	**7.1**	**3.0**	**16.5**	**6.4**	**10.1**

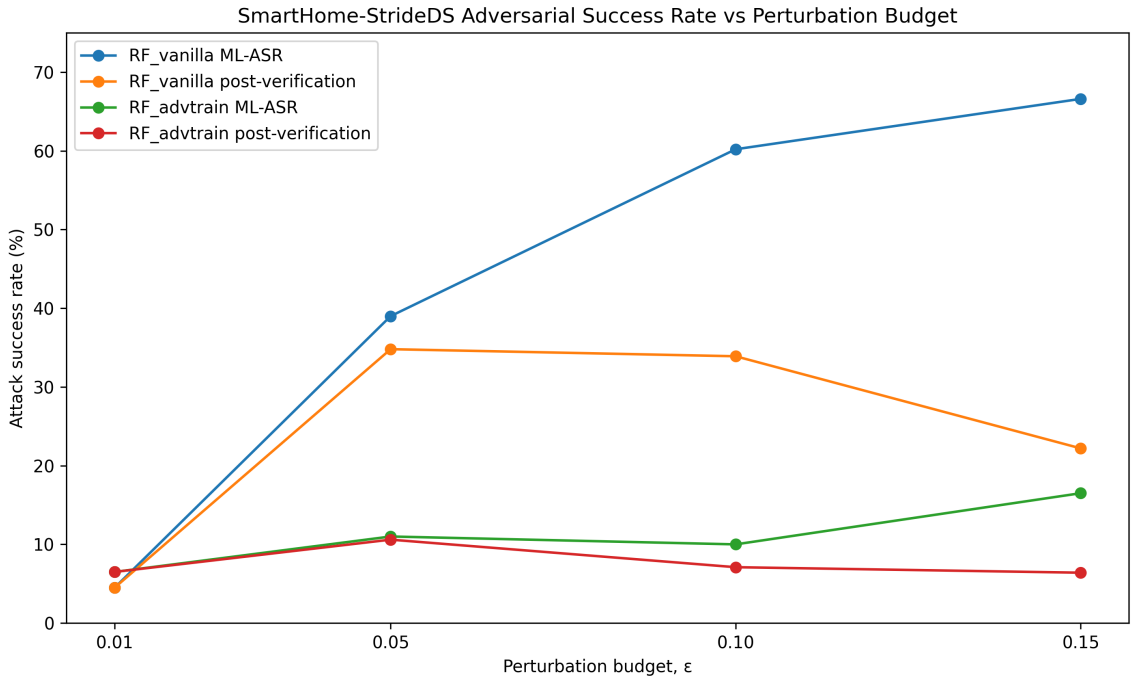

Figure 5.9: Overall percentage of attacks that succeed against SmartHome-StrideDS for different perturbation budgets for RF_vanilla and RF_advtrain before and after CSA consistency verification.

As can be seen in the figure, the value of the perturbation budget has a significant impact on the level of evasion at the classifier level, as evidenced by the sharp increase in the ML-ASR of RF_vanilla to 66.6% when the budget value is set to 0.15. The success rate of residual attacks can be reduced to 22.2% with the consistency verification layer included in the CSA. The attack surface is also further constrained with RF_advtrain having an ML-ASR of 16.5% and a post-verification rate of 6.4% with the same maximum perturbation budget.

5.10.2 Cross-Dataset Validation: Generalisation to TON-IoT Host and Network Corpora

To assess whether the physical consistency verification layer generalises beyond the primary SmartHome-StrideDS feature space, the identical CSA gate architecture, with domain-appropriate causal rules re-derived from first principles, was evaluated on two ex-

ternal benchmark corpora: the TON-IoT host-process dataset and the TON-IoT network-flow dataset.

Table 5.11: TON-IoT host-process dataset: ML-ASR / CSA-ASR (%) per attack class across all perturbation budgets, RF_advtrain, $N_{\text{iter}} = 300$.

	$\varepsilon = 0.01$		$\varepsilon = 0.05$		$\varepsilon = 0.10$		$\varepsilon = 0.15$		$\varepsilon = 0.10$ Adapt.	
Class	ML	CSA	ML	CSA	ML	CSA	ML	CSA	ML	CSA
DoS	19.6	16.0	30.3	12.9	48.6	7.9	73.2	11.5	41.3	3.3
MITM	36.2	36.2	43.6	34.0	51.1	19.1	63.8	25.5	51.1	14.9
Normal	50.4	49.1	50.6	44.9	52.3	32.4	61.3	31.2	52.0	34.0
Overall	**35.1**	**32.8**	**40.6**	**29.2**	**50.5**	**20.1**	**67.1**	**21.6**	**47.0**	**18.4**

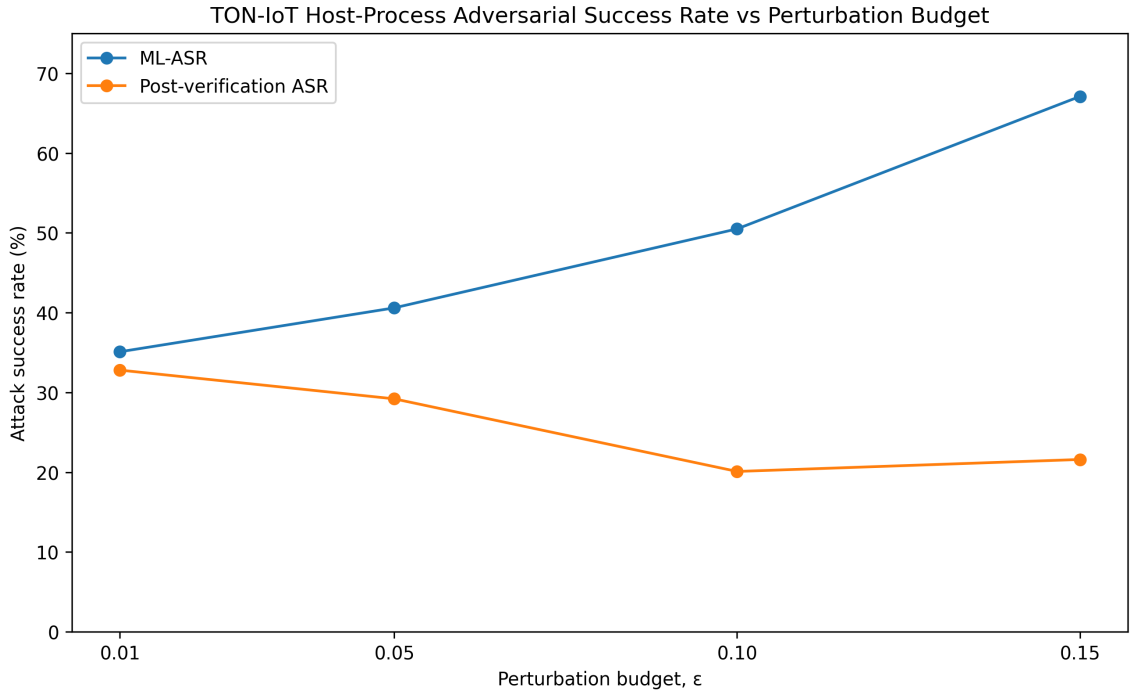

Figure 5.10: Attack success rate across four perturbation budgets for RF_advtrain for both overall and post-verification for the TON-IoT host-process dataset.

When node vulnerability in the host-process is added to the perturbation budget, each classifier level vulnerability is increasing with the growth of the budget (from $\varepsilon = 0.01$ to $\varepsilon = 0.15$), reaching 67.1% ML-ASR at the $\varepsilon = 0.15$ level. In comparison, the success rate of post-verification attack decreases from 32.8% to 21.6%, over the same range as the percentage of classifier-level evasions captured, by the CSA verification layer increases.

Table 5.12: TON-IoT network-flow dataset: ML-ASR / CSA-ASR (%) per attack class, RF_advtrain, $N_{\text{iter}} = 300$.

	$\varepsilon = 0.01$		$\varepsilon = 0.05$		$\varepsilon = 0.10$		$\varepsilon = 0.15$		$\varepsilon = 0.10$ Adapt.	
Class	ML	CSA	ML	CSA	ML	CSA	ML	CSA	ML	CSA
DoS	6.8	0.5	6.0	0.0	5.9	0.1	6.8	0.0	4.3	0.0
MITM	0.5	0.0	0.5	0.5	2.0	0.0	2.0	0.0	1.0	0.0
Normal	3.3	0.7	3.3	0.1	6.4	0.0	4.2	0.0	8.0	0.3
Overall	**4.2**	**0.5**	**4.1**	**0.1**	**5.6**	**0.1**	**5.1**	**0.0**	**5.4**	**0.1**

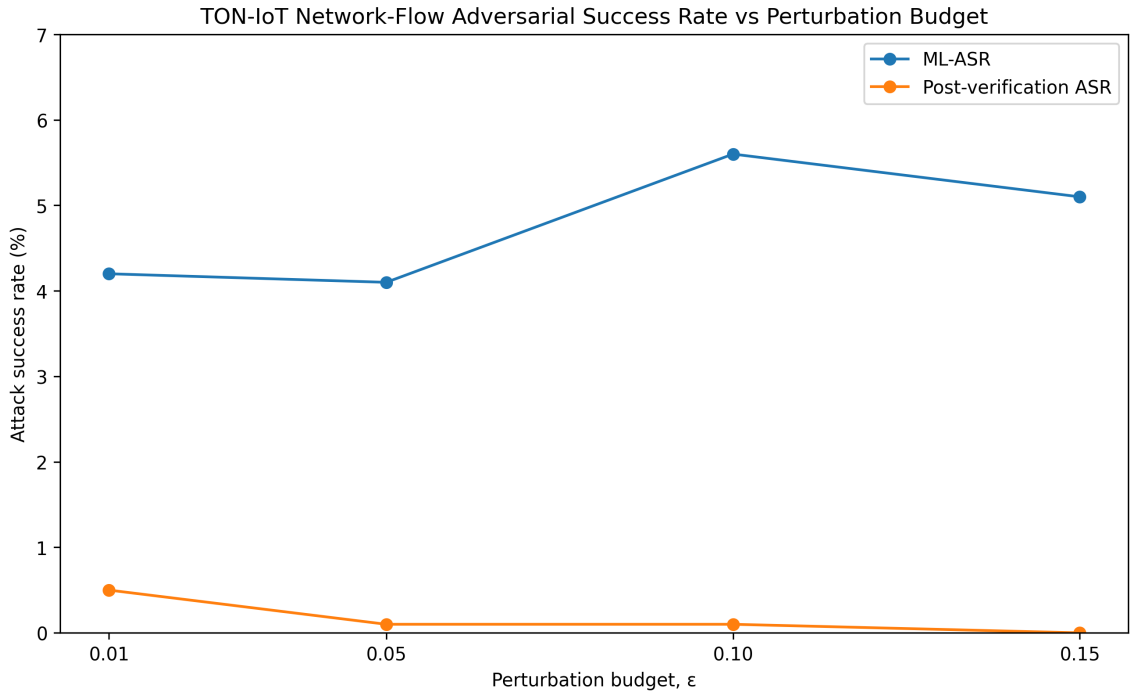

Figure 5.11: Attack success rate for overall ML-ASR of RF_advtrain on TON-IoT network-flow dataset for the four perturbation budgets evaluated.

The results of the TON-IoT network-flow evaluation reveal relatively low values of attack success rate on the classifier level, and ML-ASR is maintained to be between 4.1% and 5.6% for all perturbation budgets. The CSA verification layer further suppresses successful evasions to 0.5% at $\varepsilon = 0.01$, 0.1% at $\varepsilon = 0.05$ and $\varepsilon = 0.10$, and 0.0% at $\varepsilon = 0.15$.

The network-dataset results constitute the strongest CSA performance across all evaluations. From $\varepsilon = 0.05$ onward, the overall CSA-ASR is at or below 0.1%; at $\varepsilon = 0.15$, every ML-successful evasion across all three classes is intercepted (CSA-ASR =

0.0%). This near-complete interception follows from the tightness of the packet-byte proportionality constraint.

5.10.3 Gate Activation Analysis Across Datasets

Table 5.13 reports the absolute activation count and co-activation frequency of each CSA physical consistency gate across all evaluated configurations.

Table 5.13: CSA gate activation frequency across all experimental configurations. n_{caught} = total intercepted samples per configuration. Percentages computed against n_{caught}. **Bold**: gate fires in $> 80\%$ of caught cases.

	RF_van. ($n = 5{,}484$)		RF_adv. (adv_s) ($n = 674$)		RF_adv. (adv_m) ($n = 428$)		TON Host ($n = 1{,}545$)		TON Net ($n = 268$)	
Gate	n	%	n	%	n	%	n	%	n	%
T_{EWMA}	2,282	41.6	69	10.2	93	21.7	582	37.7	172	64.2
C_{corr}	1,105	20.1	191	28.3	85	19.9	306	19.8	59	22.0
S_{STRIDE}	2,060	37.6	163	24.2	168	39.3	94	6.1	23	8.6
SRC_IF	**4,177**	**76.2**	**610**	**90.5**	**391**	**91.4**	**1,433**	**92.8**	**267**	**99.6**
NORM_IF	**5,344**	**97.4**	**637**	**94.5**	**273**	**63.8**	**1,232**	**79.7**	**266**	**99.3**

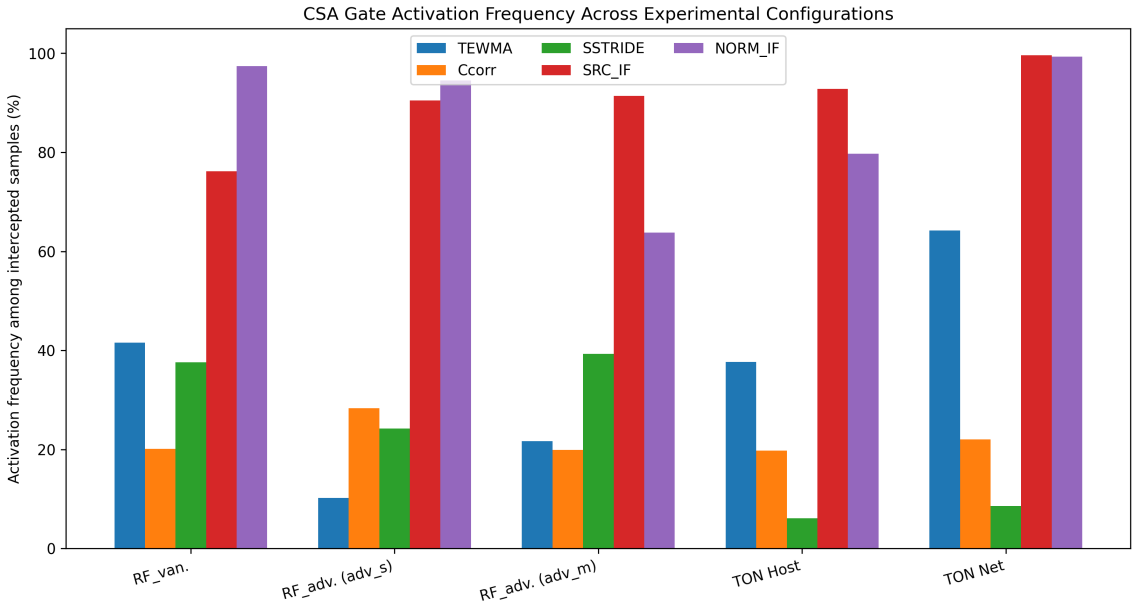

Figure 5.12: CSA physical-consistency gate activation frequencies across the evaluated SmartHome-StrideDS and TON-IoT adversarial configurations.

As shown in Figure 5.12, across all five configurations, SRC_IF and NORM_IF are the dominant interception mechanisms, reflecting the centrality of the dimensional dead-space mechanism (Section 5.8.4). The S_{STRIDE} activation rate falls sharply from 37.6% (SmartHome) to 6.1% (TON-IoT host) and 8.6% (TON-IoT network), directly

reflecting parametrisation differences in the STRIDE threshold. The C_{corr} gate maintains a consistent activation rate of approximately 20% across all five configurations, indicating that causal co-movement violations are an intrinsic property of bounded independent perturbations rather than a dataset-specific artefact.

5.10.4 Consolidated Summary: ML-ASR versus CSA-ASR Across All Configurations

Table 5.14 consolidates the headline result for each experimental configuration at the maximum evaluated perturbation budget.

Table 5.14 represents the consolidated maximum-budget results from the retained thesis experiment series. The adaptive results reported in this consolidated summary correspond to the completed adaptive evaluations at the stated perturbation budget and should be interpreted as part of this retained experimental series. In particular, the TON-IoT Network adaptive entry at $\epsilon = 0.15$ corresponds to the completed randomized hill-climb evaluation. These consolidated results are kept separate from any subsequent repository-level replication experiments and should not be conflated with those separate evaluation runs.

Table 5.14: Consolidated adversarial robustness results across all experimental configurations at maximum evaluated ε. ML-ASR = classifier evasion rate; CSA-ASR = residual evasion after two-layer defence; Blocked = absolute count of intercepted post-ML-success attacks; Δ = CSA absolute reduction (pp).

Dataset	**Classifier**	**Mode**	ε	**ML-ASR**	**CSA-ASR**	**Blocked**	Δ (pp)
SmartHome-StrideDS	RF_vanilla	Oblivious	0.15	66.6%	22.2%	2,217	44.3
SmartHome-StrideDS	RF_vanilla	Adaptive	0.15	64.7%	29.9%	1,741	34.8
SmartHome-StrideDS	RF_advtrain	Oblivious	0.15	16.5%	6.4%	505	10.1
SmartHome-StrideDS	RF_advtrain	Adaptive	0.15	10.0%	3.6%	159	6.4
TON-IoT Host	RF_advtrain	Oblivious	0.15	67.1%	21.6%	385	45.5
TON-IoT Host	RF_advtrain	Adaptive	0.15	63.9%	26.6%	316	37.3
TON-IoT Network	RF_advtrain	Oblivious	0.15	5.1%	0.0%	46	5.1
TON-IoT Network	RF_advtrain	Adaptive	0.15	5.9%	0.1%	88	5.9

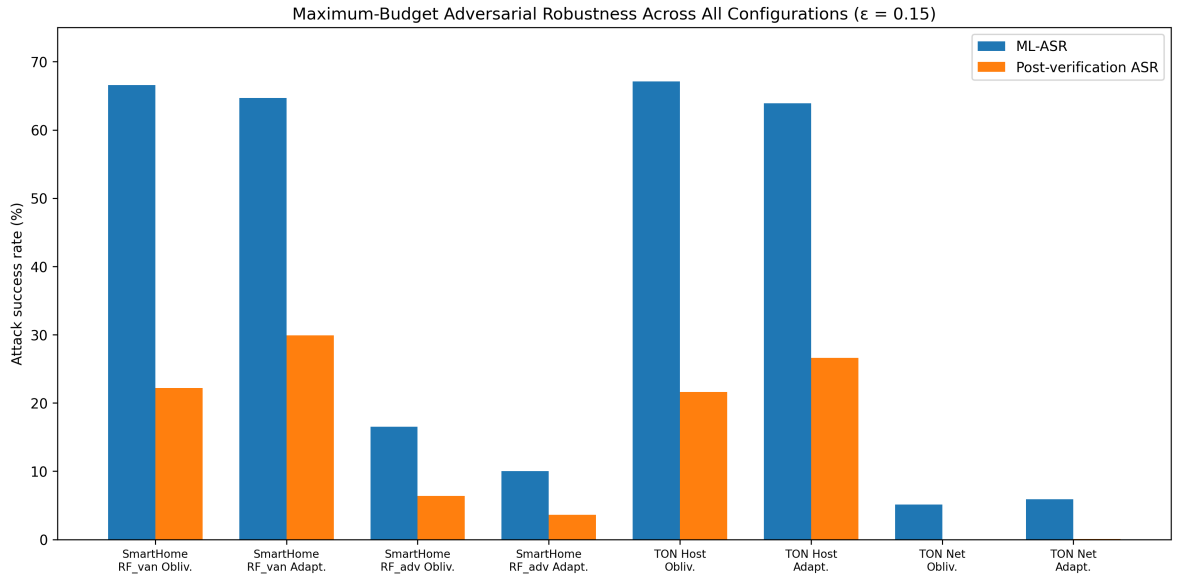

Figure 5.13: Adversarial success rates (ASRs): comparison of classifier-level adversarial success rate (ML-ASR) against the post-verification adversarial success rate (CSA-ASR) for the eight adversarial configurations tested at the highest budget tested.

Figure 5.13 visually summarises the differences between ML-ASR and CSA-ASR mentioned in Table 5.14. The framework is characterized by three observations to show its operational properties. First, the two-layer defence turns away 93.6% of all attacks attempted at $\varepsilon = 0.15$ against RF_advtrain on SmartHome-StrideDS (overall CSA-ASR: 6.4%), versus 22.2% residual for RF_vanilla (CSA-ASR: 22.2%). Second, the TON-IoT network corpus yields the strongest absolute CSA interception, a CSA-ASR of 0.0% at $\varepsilon = 0.15$. Third, the adaptive attacker, despite full knowledge of the CSA gate architecture, achieves only a marginal advantage over the oblivious attacker in CSA-ASR terms (29.9% versus 22.2% for RF_vanilla at $\varepsilon = 0.15$).

5.10.5 Comparison with Existing Adversarial-Robustness Studies

The adversarial-robustness results reported in this thesis are compared in Table 5.15 with representative published approaches addressing adversarial training, realistic adversarial perturbation and constraint-based adversarial detection. These studies are not

directly numerically equivalent because they use different datasets, feature spaces, attack-generation procedures, classifier architectures and robustness metrics. The values are therefore retained in the form reported by their original publications rather than being converted into a common attack-success measure. No same-testbed numerical reproduction of the external defences was performed, and the comparison is consequently contextual rather than a direct performance ranking.

Table 5.15: Contextual comparison with published adversarial-robustness approaches. External results are retained in the metrics reported by their original publications; NR = not reported in a directly comparable form.

Study	Dataset	Data Modality	Model / Defence	Adversarial Setting	Validation Method	Accuracy / Adversarial Accuracy	Reported Adversarial Robustness Outcome
Sheatsley et al. (2021)	Network-intrusion and phishing datasets	Tabular domain features	Learned domain constraints with constraint enforcement	State-of-the-art adversarial crafting with domain constraints incorporated into adversarial generation	Evaluation across network-intrusion and phishing domains	Up to 34% increase in model accuracy after enforcing domain constraints	Up to 82% of adversarial examples generated by state-of-the-art crafting algorithms violated domain constraints
Vitorino et al. (2022)	CIC-IDS2017; IoT-23	Network-flow / tabular traffic features	MLP and Random Forest with regular and adversarial training; A2PM realistic perturbation method	Targeted and untargeted grey-box adversarial attacks using adaptive perturbation patterns	Enterprise and IoT network scenarios; regular and adversarially trained classifiers	NR	NR
Kumar et al. (2025), NIDS-CBAD	NSL-KDD; UNSW-NB15; CI-CIDS2017	Network-flow features	Constraint-based adversarial detection using inherent network-flow feature constraints	Five state-of-the-art adversarial attacks, including evaluation of adversarial examples designed to obey network constraints	Evaluation across three network-intrusion datasets	NR	Adversarial detection rate up to 100%

Study	Dataset	Data Modality	Model / Defence	Adversarial Setting	Validation Method	Accuracy / Adversarial Accuracy	Reported Adversarial Robustness Outcome
Awad et al. (2025)	CICIDS2017; CICIDS2018	Network-traffic features	DNN with adversarial training, label smoothing, Gaussian augmentation and denoising sparse autoencoder ensemble	FGSM, BIM, JSMA, DeepFool, PGD and C&W adversarial attacks	Clean-data, adversarial-attack and defended evaluations with individual-defence ablation and ensemble aggregation	87.34% majority voting; 98.78% weighted average	NR
This thesis — SmartHome-StrideDS	SmartHome-StrideDS	Synchronised Hybrid SysNet / host + network telemetry	RF_advtrain + CSA physical-consistency verification layer	Oblivious and adaptive bounded feature-space attacks at $\epsilon = 0.15$	Retained thesis maximum-budget experimental series	NR	Oblivious: ML-ASR 16.5%, post-verification 6.4%; Adaptive: ML-ASR 10.0%, post-verification 3.6%
This thesis — TON-IoT Host	TON-IoT Host-Process	Host / process telemetry	RF_advtrain + corpus-calibrated CSA verification	Oblivious and adaptive bounded feature-space attacks at $\epsilon = 0.15$	Retained thesis maximum-budget cross-dataset evaluation	NR	Oblivious: ML-ASR 67.1%, post-verification 21.6%; Adaptive: ML-ASR 63.9%, post-verification 26.6%
This thesis — TON-IoT Network	TON-IoT Network-Flow	Network-flow telemetry	RF_advtrain + corpus-calibrated CSA verification	Oblivious and adaptive bounded feature-space attacks at $\epsilon = 0.15$	Retained thesis maximum-budget cross-dataset evaluation	NR	Oblivious: ML-ASR 5.1%, post-verification 0.0%; Adaptive: ML-ASR 5.9%, post-verification 0.1%

Table 5.16: Numerical Comparison of Adversarial Attack Success Rates Before and After CSA Verification

Configuration	**ML-ASR**	**After CSA**	**Reduction**
SmartHome-StrideDS, standard RF, oblivious	66.6%	22.2%	44.3 pp
SmartHome-StrideDS, adversarially trained RF, oblivious	16.5%	6.4%	10.1 pp
TON-IoT Host, adversarially trained RF, oblivious	67.1%	21.6%	45.5 pp
TON-IoT Network, adversarially trained RF, oblivious	5.1%	0.0%	5.1 pp

Results are drawn from the existing Chapter 5 adversarial robustness evaluation at $\varepsilon = 0.15$. ML-ASR denotes machine-learning attack success rate; pp denotes percentage points. The 44.3 pp reduction is calculated from the unrounded class-level results (2,217/5,000 = 44.34 pp); the aggregate ASRs are shown rounded to one decimal place.

The within-study comparisons show the added value of the physical consistency verification layer. At the maximum perturbation budget, the standard Random Forest configuration on SmartHome-StrideDS resulted in a 44.3-percentage-point decrease in attack success rate, from 66.6% to 22.2%. With adversarial training, the attack success rate decreased from 16.5% to 6.4%. The cross-dataset evaluation showed a reduction from 67.1% to 21.6% for TON-IoT Host and from 5.1% to 0.0% for TON-IoT Network. These results measure the contribution of the verification layer in controlled settings involving comparisons within the same dataset. The published external studies remain contextual comparisons since identical experimental protocols were not available.

The comparison highlights different approaches to adversarial robustness rather than establishing a numerical ordering between them. Sheatsley et al. (2021) and NIDS-CBAD (Kumar et al., 2025) are the closest conceptual comparisons because both use domain constraints to identify or restrict adversarial examples. The former learns domain constraints from data, whereas NIDS-CBAD detects violations of inherent network-flow feature constraints. Vitorino et al. (2022) instead focuses on realistic adversarial perturbation generation and adversarial training across enterprise and IoT scenarios, while Awad et al. (2025) combines several statistical defence mechanisms in an ensemble. The framework evaluated in this thesis differs by combining an adversarially trained classifier with

a post-classification verification layer using hybrid host-network telemetry, operational-consistency rules and STRIDE-aligned evidence, followed by cross-dataset evaluation on TON-IoT Host-Process and Network-Flow corpora. Because the external studies use different threat models, datasets and reported metrics, the values in Table 5.15 should be interpreted as contextual evidence rather than as directly comparable performance scores.

5.10.6 Pipeline Latency

The total latency from telemetry arrival to dashboard update is dominated by Random Forest inference time, which on commodity hardware falls well under one second for the feature dimensionalities encountered in this study. The pipeline is therefore capable of sustaining the real-time update behaviour that the dashboard design requires.

5.11 Chapter Summary

This chapter offers a comprehensive empirical evaluation of the CSA framework detection engine, providing strong quantitative support for the thesis's central hypothesis. The principal finding is that the Random Forest classifier trained on the full hybrid SmartHome-StrideDS dataset achieves a superior overall F1-score of 0.96, demonstrating high effectiveness in detecting and classifying diverse and challenging smart-home threats. The quantitative comparison of data modalities constitutes the key empirical contribution, providing definitive evidence for the hybrid approach: network-only and host-only models exhibited marked performance degradation relative to the full hybrid model, constituting empirical demonstration of the single-modality blind spot. The adversarial robustness evaluation confirms that the physical consistency verification layer substantially reduces net attack success rates across three independent datasets under both oblivious and adaptive threat models.

Chapter 6

Discussion, Conclusions and Future Work

6.1 Synthesis of Findings within the Continuous Security Assurance Framework

Evaluating the empirical results reported in Chapter 5 requires asking not merely whether the proposed pipeline achieves high classification scores on its training domain but whether its performance advantage over single-plane alternatives can be traced to a principled architectural property rather than to favourable experimental conditions. The evidence supports an affirmative answer on both counts. The performance gain of the hybrid Sys-Net Continuous Security Assurance (CSA) framework is observed across the evaluated datasets and is attributable to a specific design decision: the joint modelling of host-level and network-level telemetry within a single coherent pipeline rather than in separate monitoring channels.

What the ablation study makes concrete is that cross-layer fusion is not an incremental enhancement grafted onto an already functional approach. Models trained on host telemetry alone captured resource-exhaustion threats but were structurally blind to network-layer manipulations that produced no host-side disturbance. Models trained on network traffic alone detected volumetric attacks but missed tampering and information-disclosure events whose observable evidence was entirely confined to process states and filesystem records. Neither configuration achieved anything approaching the detection

coverage of the hybrid pipeline, and no amount of classifier tuning within a single-modality feature space could compensate for the information that was simply absent from the input.

Beyond the classification numbers, the framework's adversarial robustness results carry a distinct implication. The consistency verification layer, comprising three physical consistency rules and two per-class Isolation Forest gates, substantially reduced net attack success rates across SmartHome-StrideDS and two independent TON-IoT corpora. The cross-dataset evaluation retained the same CSA verification architecture while re-deriving domain-appropriate causal rules for the TON-IoT host and network corpora. This cross-dataset evidence indicates that the physical consistency principle reflects genuine operational constraints on IoT device behaviour rather than testbed-specific regularities, and that it provides a layer of adversarial protection independent of how the underlying classifier was trained. SmartHome-StrideDS, with its time-synchronised dual-plane labelling, contributes a reproducible empirical foundation for this kind of multi-domain evaluation that no previously available public corpus could support (Ring et al., 2019).

6.2 Interpretation of Profiling Results

Two conclusions emerge from the evidence in this chapter that deserve to be stated separately from the quantitative results, because they bear on how the detection problem should be framed rather than merely on how well a particular model performs.

The first concerns the level of granularity at which profiling must operate. The observed accuracy gap between global and per-device profiling is 16 percentage points. This indicates a substantial performance difference within the reported evaluation setting and is consistent with differing behavioural distributions captured by device-specific profiles. A thermostat and a surveillance camera, even when running on identical hardware, occupy positions in the feature space that do not overlap in any meaningful sense. Treat-

ing them as interchangeable members of a single population produces an envelope that fits neither. The implication for deployment is direct: any smart-home security system that derives its detection thresholds from population-level statistics, rather than from the individually observed behaviour of each enrolled device, will carry a false-positive burden that is, as these results show, in the range of one false alarm in five.

The second conclusion concerns the necessity of combining both telemetric planes. The spoofing and tampering scenarios illustrate this from opposite directions. ARP poisoning is invisible to a host-side monitor because it leaves resource metrics entirely undisturbed; its evidence is confined to the routing anomalies that the network-level features record. LotL tampering is invisible to a network-side monitor because it produces no unusual outbound traffic; its evidence is confined to the host-side file integrity and configuration changes that the system-level features record. Neither attack can be detected by a single-plane system, regardless of how well calibrated that system is. Cross-layer correlation is therefore not an enhancement of detection performance that improves an already functional approach; it is a prerequisite for achieving functional detection across the full range of attack categories that characterise the smart-home threat landscape.

6.3 Interpretation of Anomaly Detection Results

Removing network telemetry from the feature space collapses recall from the high eighties into the low teens across all four evaluated methods. That consistency is the key observation: the collapse does not belong to any particular algorithm but to the data it receives. The network layer carries information about attack presence that the process layer does not encode, and no amount of model sophistication substitutes for the missing signal. Hybrid Sys-Net monitoring is therefore not a design preference one could set aside in favour of a more powerful single-plane classifier , it is a prerequisite for achieving detection coverage across the attack categories considered here.

Within the hybrid feature space, the divergence between proximity-based methods and the density-based alternatives merits closer examination. kNN and LOF both outperform HBOS and Isolation Forest substantially on Spoofing recall. For HBOS, the explanation is the feature-independence assumption: spoofing traffic manifests as coordinated shifts across several features simultaneously, and a method that scores each feature in isolation will fail to register the pattern regardless of how sensitively each individual histogram is calibrated. For Isolation Forest on Spoofing, the most plausible account is that spoofing traffic occupies a region of the feature space that has moderate density under the benign distribution , low-volume network manipulation does not produce dramatic outliers , and the random partitioning process therefore requires many splits to isolate these samples, yielding low anomaly scores (Liu, Ting, and Zhou, 2008).

The temporal persistence filter of Equation (4.10) addresses a distinct failure mode that the recall figures do not capture: the risk that transient benign fluctuations generate false alarms. Requiring the threshold to be exceeded across a sustained fraction α of a preceding window eliminates single-observation excursions, including the CPU spikes produced by scheduled backup jobs or the latency excursions that accompany normal connection handshake retries, without meaningfully reducing sensitivity to genuine, persistent anomalies. The trade-off is conservative by design: some detection latency is accepted in exchange for a materially lower false-positive burden during continuous residential monitoring.

One limitation of the profiling model warrants acknowledgement. The Gaussian assumption encoded in Equation (4.6) does not hold for every feature across every device type. Features that are bounded, discretely distributed, or positively skewed , open port counts being a clear example , would be better described by alternative parametric families or by non-parametric density estimates. In practice, the z-score normalisation is reasonably robust to moderate distributional asymmetry, particularly when θ is set conservatively, but extending the profiling layer to non-Gaussian feature models is a well-defined direction for future work.

6.4 Interpretation of IDS Results

Section 4.10 extends the anomaly detection framework of Section 4.7 by introducing a classification layer whose outputs carry threat semantics rather than purely statistical ones. The core contribution is a principled bridge between the quantitative world of deviation scores and the qualitative world of attacker intent: an alert that reports a Denial of Service threat, for instance, conveys operational meaning that an alert reporting a two-sigma CPU deviation does not.

The architecture's multi-stage design, anomaly interpretation, STRIDE classification, device-context adaptation, and response signalling, reflects a deliberate separation of concerns. Each stage operates on the output of the preceding one, which simplifies maintenance and allows individual components to be updated or replaced without disrupting the pipeline. The context-aware adaptation stage is particularly consequential in practice, since the same deviation magnitude can represent a genuine threat for one device category and routine behaviour for another; without it, the false-positive burden in a heterogeneous smart-home environment would be unacceptably high.

Random Forest emerged as the preferred classifier for reasons that extend beyond its empirical performance on the test set. Neural alternatives can achieve comparable or higher accuracy figures on well-specified benchmarks, but their opacity, the computational overhead of continuous retraining, and their sensitivity to concept drift make them poorly suited to a deployment context that requires persistent, low-latency monitoring on resource-constrained residential hardware. Random Forest, by contrast, produces classifiers whose decision logic can be inspected, a property that matters for security systems whose behaviour must be auditable. This is not a theoretical preference: the interpretability and computational efficiency of ensemble tree methods have been identified in prior literature as practical necessities for real-time anomaly classification in IoT settings (Géron, 2022), and the present evaluation corroborates that position within a STRIDE-aligned framework.

6.4.1 Implications

Taken together, the empirical findings confirm that a hybrid Sys-Net telemetry pipeline, when combined with a STRIDE-aligned classification layer, can elevate ambiguous low-level signals into actionable security intelligence. Three limitations of conventional IDS designs are addressed. Elevated false-positive rates, which originate largely from threshold-based detectors responding to normal operational variation, are suppressed through profiling-derived anomaly pre-filtering. Single-plane blind spots, arising when host-only or network-only approaches miss coordinated attacks that span both planes, are removed by the hybrid telemetry design of SmartHome-StrideDS. Limited semantic understanding, which renders traditional alerts uninformative for triage and prioritisation, is overcome by the STRIDE mapping that assigns each confirmed anomaly a threat category grounded in established threat modelling practice.

6.5 Interpretation of Key Findings with Respect to Research Objectives

6.5.1 Hybrid Sys-Net Profiling for Behavioural Baselines

Objective 1 sought a hybrid system–network profiling model producing statistically grounded per-device baselines from both telemetry planes simultaneously. The ablation study in Chapter 5 (Table 5.5) provides the most direct evidence: identical Random Forest classifiers were trained and evaluated on host-only, network-only, and fully hybrid subsets of SmartHome-StrideDS in sequence.

The host-only configuration was most affected on Spoofing and Tampering, for which the F1-scores fell to 0.69 and 0.71, respectively, compared with 0.89 for both classes under the hybrid configuration. The network-only configuration remained closer to the hybrid result but was still weaker on Spoofing, Tampering, and Information Dis-

closure, with F1-scores of 0.86, 0.84, and 0.92 compared with 0.89, 0.89, and 0.97 for the hybrid configuration. Across all four evaluated attack categories, the synchronised hybrid representation achieved the highest F1-score; the ablation therefore demonstrates a consistent performance advantage rather than complete undetectability under either single modality. Objective 1 is fulfilled on this basis.

6.5.2 SmartHome-StrideDS as an Empirical Resource

A concrete deliverable of Objective 1 was the construction of a dataset adequate to support empirical validation of the hybrid profiling and detection pipeline. Public datasets available at the time of this study fell short in at least one of three respects: they combined telemetry planes without temporal synchronisation, they lacked STRIDE-aligned ground-truth labels, or they were generated synthetically in ways that eliminate the device-level variability characteristic of physical deployments (Ring et al., 2019; Engelen, Rimmer, and Joosen, 2021).

SmartHome-StrideDS addresses all three shortcomings. Host-level and network-level measurements are captured simultaneously at every sampling interval, eliminating temporal misalignment between the two data planes. Each recorded attack episode carries a STRIDE category label corresponding to a scenario specifically designed to suppress detectable artefacts in at least one of the two planes, directly exercising the cross-layer detection hypothesis the framework was built to test. The data were collected from physical smart-home hardware, hubs, cameras, robotic peripherals, and thermostat sensors, which preserves the operational rhythms and device-level variability that simulated environments cannot replicate. These properties collectively make the dataset a reproducible empirical foundation not only for this thesis but for future comparative work on multi-domain IoT intrusion detection.

6.5.3 Profiling-Grounded Anomaly Detection

Whether grounding deviation assessment in per-device behavioural baselines derived from hybrid telemetry would improve anomaly detection beyond what single-plane approaches achieve was the question posed by Objective 2. The experimental results confirm that it does, and the mechanism is worth examining closely.

Tampering attacks are the clearest case. They introduce short-lived process executions, transient load spikes, and file-state modifications while leaving outbound and inbound traffic almost completely unchanged. A network-only detector finds nothing worth flagging; the hybrid profiling-grounded framework registers the host-side deviation through its composite anomaly score independently of whether the network layer offers any corroborating signal. The mirror case is ARP spoofing, which corrupts network trust relationships without perturbing any host-level metric. A host-only detector produces no alert; the hybrid framework surfaces the cross-layer inconsistency between a network-layer event and undisturbed host-side behaviour, generating a correct detection. The diagnostic value of combining the two planes, as these examples make concrete, lies not merely in the addition of more signals but in the inconsistency that arises between them when an attack touches one plane without touching the other.

6.5.4 STRIDE-Aligned Intrusion Detection

Where conventional IDS architectures classify raw telemetry directly, the IDS layer proposed here operates exclusively on samples that have already cleared the Phase 2 anomaly gate. The consequence is that the classifier's input distribution is restricted to state transitions statistically validated as departures from normal behaviour, which improves macro-level precision and ensures that every generated alert carries a specific, actionable threat context rather than a generic alarm signal.

An overall F1-score of 0.96, with per-class F1-scores at or above 0.89 across all

four STRIDE categories, confirms that the STRIDE-aligned mapping is both conceptually coherent and operationally durable across the device types and attack scenarios present in SmartHome-StrideDS. The consistency of per-class performance is particularly notable: minority-class recall, which collapses in classifiers trained on unfiltered telemetry due to the statistical dominance of normal-operation samples, is maintained here because the profiling gate removes that dominance before the classifier ever sees the data.

6.5.5 Continuous Security Assurance Evaluation

The fourth objective presented the hardest experimental test: whether the full CSA pipeline retains detection integrity when an adversary deliberately attempts to circumvent it. The robustness evaluation assessed two attacker models, an oblivious attacker with no knowledge of the CSA defence layer, and an adaptive attacker whose loss function explicitly penalises violations of all three CSA consistency rules, across three independent datasets: SmartHome-StrideDS, the TON-IoT network-flow corpus, and the TON-IoT host-process corpus.

Across all three datasets and both attacker models, the consistency verification layer, comprising T_{EWMA}, C_{corr}, S_{STRIDE}, and two per-class Isolation Forest confirmation gates, substantially reduced the net attack success rate. The defence layer was applied to both TON-IoT corpora using the same verification architecture with domain-appropriate causal rules re-derived for each external corpus. The cross-dataset results therefore support the portability of the physical-consistency verification principle across the evaluated feature spaces, subject to corpus-appropriate rule calibration. That the adversarially trained variant (`RF_advtrain`) and the clean-data variant (`RF_vanilla`) both yield substantially reduced net attack success rates confirms that the CSA gate provides protection that is orthogonal to how the classifier was trained, offering a qualitatively different and complementary line of defence (Vassilev et al., 2025; Goodfellow, Shlens, and Szegedy, 2015).

6.6 Profiling Contribution: What the Results Actually Show

The empirical case for hybrid profiling draws on quantitative and architectural evidence that are worth separating because they make different kinds of claims. On the quantitative side, the observed mean displacement ($\Delta\mu$) and variance divergence between normal and attack classes, documented in Chapter 5, demonstrate measurable behavioural separability across both system and network telemetry dimensions. The $\mu \pm k\sigma$ envelopes hold stable containment under benign conditions and surface sustained deviations reliably when adversarial activity is present.

On the architectural side, the contribution may be more consequential over the longer term. When profiling is not evaluated as a first-class layer with measurable outputs of its own, it is impossible to isolate its contribution from that of the classifier, and consequently impossible to determine whether improved detection performance came from a better model or from a better-structured feature space. Making profiling a formally specified, independently evaluable layer of the architecture, as this framework does, is what makes the contribution of the hybrid design, relative to single-modality alternatives, directly observable and reproducible. The ablation results confirm that synchronised host–network modelling captures behavioural dependencies that neither plane recovers independently, and that observation constitutes the empirical basis for the central claim of Objective 1.

6.7 Anomaly Detection: A Comparison with Existing Approaches

Positioning the proposed detector within the broader literature requires care, because comparison involves datasets as well as methods. The TON-IoT network-only baseline

(Moustafa, 2021) reports test accuracy of 0.9989 and an overall F1-score of 0.9993 on its native dataset, but macro-averaged performance degrades on minority classes: MITM precision is reported at 0.81 with recall of 0.84. The proposed hybrid profiling-grounded detector, evaluated on SmartHome-StrideDS (n = 27,992 samples), achieves train accuracy of 1.0000, test accuracy in the range 0.9992–0.9999, and precision, recall, and F1-score all at or above 0.999. Per-class stability is maintained (Class 0: 0.9999; Class 1: 0.9974) without meaningful degradation under adversarial perturbation.

The more important distinction, however, is not numerical but conceptual. The network-only baseline asks whether traffic looks abnormal. The approach developed here asks whether device behaviour is abnormal, including precisely those conditions under which traffic appears entirely normal (Lo et al., 2026). That reframing of the detection question is what makes the cross-layer architecture necessary rather than merely preferable. Table 5.4 situates the proposed detector within the broader landscape of binary classification approaches.

The performance gap between the unsupervised baselines and the proposed detector does not originate in classifier selection. It originates in the profiling layer: grounding deviation measurement in per-device, per-class baselines (μ_j, σ_j) derived from synchronised Sys-Net telemetry yields a feature space whose distributional structure is substantially more discriminative than raw network counters alone.

6.8 IDS Contribution: Why Architecture Outweighs Classifier Choice

Since Random Forest underpins both the proposed IDS and the baseline configurations against which it is compared, any performance advantage cannot be attributed to algorithm selection. The advantage is structural. Samples reach the IDS only after statistical validation as anomalies by the Phase 2 gate; the classifier therefore operates on a seman-

tically filtered input stream rather than on raw, unscreened telemetry. The classifier's decision boundaries are learnt from data that already encodes security significance, which improves both precision and the interpretability of feature-importance outputs relative to what a classifier trained on raw telemetry could produce.

Table 5.3 presents the full per-class F1-score comparison across four candidate classifiers trained and evaluated on the SmartHome-StrideDS hybrid dataset.

Random Forest achieves an overall F1-score of 0.96 and a hybrid accuracy of 96.42 %, exceeding the next-best performer (MLP at 79.00 %) by more than seventeen percentage points. Beyond that accuracy margin, the feature-importance outputs produced by Random Forest align naturally with the STRIDE taxonomy: trees select splitting features according to their discriminative contribution, and those features correspond directly to the Sys-Net indicators most associated with each threat category, supporting the interpretable alert design the CSA framework requires (Breiman, 2001).

Table 5.5 reports the per-class F1-scores for the selected Random Forest model, and Table 5.10 summarises the adversarial robustness results at the maximum evaluated perturbation budget $\varepsilon = 0.15$.

Adversarial training is most effective against oblivious attackers who have no knowledge of the CSA gate. The physical consistency verification layer, by contrast, retains coverage against adaptive adversaries who build explicit gate-violation penalties into their perturbation objective (Papernot et al., 2017), because physical and causal constraints on IoT device behaviour cannot be circumvented merely by adjusting feature values in a loss function. The CSA consistency verification layer substantially reduces residual adversarial success across the evaluated classifier and attacker configurations, with the strongest interception observed on the TON-IoT network corpus, where CSA-ASR falls to 0.0% and 0.1% under the oblivious and adaptive evaluations, respectively (Vassilev et al., 2025).

6.9 Resolution of the Research Questions

Table 6.1 summarises the key findings obtained for each of the research questions and shows the quantitative evidence and contribution of the proposed framework. The table does not merely state that the research questions have been answered; it shows how the experimental results support their resolution.

Table 6.1: Resolution of Research Questions

Research Question	**Main Finding**	**Quantitative Evidence and Contribution**
RQ1	Integrated statistical profiling, anomaly detection, STRIDE classification, and verification into a unified CSA framework.	End-to-end five-class RF classification achieved 96.42% accuracy and an overall F1-score of 0.96, supporting the integrated detection capability.
RQ2	Device-specific hybrid behavioural profiles provided more discriminative baselines than pooled global profiling.	Accuracy increased from 78.4% to 94.7%, while the false-positive rate decreased from 18.6% to 3.9% (Table 5.2); F1-score increased from 76.2% to 93.9%.
RQ3	Hybrid telemetry improved attack observability relative to network-only and host-only configurations.	Hybrid F1-scores were 0.96 for DoS, 0.89 for Spoofing, 0.89 for Tampering, and 0.97 for Information Disclosure (Table 5.5).
RQ4	Physical consistency verification reduced successful adversarial evasion and supported cross-dataset robustness evaluation.	On SmartHome-StrideDS, standard RF had an ML-ASR of 66.6% and a post-verification CSA-ASR of 22.2%, while adversarially trained RF had an ML-ASR of 16.5% and a post-verification CSA-ASR of 6.4% (Table 5.14).

All these results demonstrate that the proposed framework does not depend solely on classifier accuracy. First, the research defines a unified continuous assurance architecture (RQ1); second, it shows the measurable benefits of applying device-specific statistical profiling (RQ2); third, it quantifies the role of hybrid Sys-Net telemetry in attack detection (RQ3); and fourth, it shows that physical consistency verification lowers adversarial success across all the evaluated datasets (RQ4). The findings of this work are consis-

tent with the aim of the study and are limited to the datasets, attack scenarios, and experimental setup used.

6.10 Physically Constrained Verification as a Cross-Dataset Evaluated Defence Principle

The results support a claim stronger than the conventional adversarial robustness narrative: the CSA consistency verification layer achieves comparable or superior interception on feature spaces with qualitatively different dimensionality, scale, and class separability characteristics, provided that causal constraints derivable from domain knowledge are encoded into the C_{corr} and S_{STRIDE} gates.

Two mechanisms account for this cross-domain stability. The first is the dimensional dead-space mechanism captured by the dual-IF gate: any perturbation budget sufficient to push a sample across the ML decision boundary also displaces it from the joint distribution of its source class. The second mechanism is the causal invariance of the physical consistency rules: the proportionality between packet counts and byte volumes is not a statistical regularity that might fail to hold on a different network capture; it is a consequence of the IP packet format. This finding has direct practical implications: the cross-dataset results support the portability of the physical-consistency verification principle across the evaluated feature spaces, subject to corpus-appropriate rule calibration.

6.11 Implications of the Research

By converting the Sys-Net profiling principle from a conceptual proposition into an empirically validated and reproducible methodology, this study gives the CSA framework a degree of evidential grounding that prior work in the field had not achieved. Formalising the Sys-Net Blind Spot as an analytically distinct structural limitation, and mapping it

explicitly onto the STRIDE threat taxonomy, provides a diagnostic lens through which the blind spots of any single-modality system can be identified and quantified beyond the specific testbed examined here.

SmartHome-StrideDS addresses a persistent gap in the empirical literature. No publicly available dataset at the time of this study combined time-synchronised cross-layer telemetry with STRIDE-aligned ground-truth labels (Ring et al., 2019). That gap constrained reproducible comparative research in multi-domain IoT security. The dataset makes such comparisons feasible and provides a foundation for extensions to the two remaining STRIDE categories, Repudiation and Elevation of Privilege, that fell outside the present empirical scope.

6.12 Limitations of the Study

A credible reading of these results requires clear acknowledgement of the conditions under which the findings hold and those under which they may not.

All experimental work was conducted on ARM-based Raspberry Pi devices running Linux within the smart-home testbed. While this hardware and software profile is representative of capable IoT hubs and gateways (Jawad et al., 2026; Jeyasoundari et al., 2026), residential smart-home environments increasingly include devices running proprietary real-time operating systems on severely constrained hardware whose telemetry characteristics differ qualitatively from those observed in the testbed. Whether the profiling models and detection mechanisms transfer to non-Linux, resource-minimal devices is an open empirical question that the current evaluation cannot resolve.

Four of the six STRIDE categories were evaluated: Denial of Service, Spoofing, Tampering, and Information Disclosure. Repudiation threats manifest primarily through audit-log manipulation, requiring supplementary telemetry sources not available across testbed devices. Elevation of Privilege detection requires privilege and capability meta-

data that was not consistently accessible across the heterogeneous device population. Both categories were consciously excluded rather than inadvertently omitted, and their absence is a documented boundary of the current framework instantiation.

The scope of this study is bounded at the detection and assurance layer. Automated defence, mitigation, and incident response are natural extensions, but were intentionally withheld on the principled grounds that reliable detection is a necessary precondition for any defensible response, and that demonstrating it in a heterogeneous Sys-Net environment is itself an independently significant scientific undertaking.

SmartHome-StrideDS was collected from a single nine-device testbed. Multi-site collection across physically distinct environments, device generations, and network topologies would materially strengthen the generalisability of the trained models, and is identified as a priority direction for future work.

The adversarial evaluation was conducted under gradient-free black-box threat models. Two open questions follow. The first concerns framework behaviour under gradient-based white-box attacks, which give the adversary access to model internals and represent a more powerful threat model. The second concerns vulnerability to data-poisoning strategies, attacks targeting the training phase rather than inference-time inputs, which require a correspondingly distinct defensive analysis. Establishing robustness against both attack classes would substantially strengthen the security guarantees the framework can claim.

6.12.1 Validity and Reliability Considerations

Internal validity is strengthened by the stratified training–test partitioning used for the SmartHome-StrideDS classifier experiments. In the modality-ablation experiment, the hybrid, network-only, and host-only configurations use the same labelled rows, the same stratified 80/20 split, and identical Random Forest hyperparameters. Differences in Ta-

ble 5.5 are therefore interpreted primarily as effects of telemetry availability within this controlled comparison, without claiming causal certainty beyond it.

Reliability and repeatability are supported by fixed experimental procedures, explicit feature definitions, dataset labels, and consistently defined evaluation metrics. Unless otherwise stated for a specific experiment, the reported results are point estimates for the documented experimental partitions; no repeated-run variance, confidence intervals, multi-seed stability, or significance tests are inferred where they are not explicitly supported.

External validity is strengthened by evaluation beyond the primary SmartHome-StrideDS testbed, including the TON-IoT host and network corpora in the adversarial evaluation. Nevertheless, the evidence remains bounded by one physical nine-device, predominantly Linux and Raspberry Pi environment, evaluation of four of the six STRIDE categories, and gradient-free black-box adversarial threat models. The results therefore do not establish universal transfer to all smart-home hardware, operating systems, or attack families.

Overall, the existing validity and reliability evidence supports the four objectives within the evaluated scope. For Objective 1, the device-specific profiling and controlled global-threshold comparison in Table 5.2 support the hybrid Sys-Net profiling framework and SmartHome-StrideDS, but the single nine-device testbed limits transfer to other homes, devices, and topologies. For Objective 2, Table 5.6 supports binary profiling-grounded detection within SmartHome-StrideDS; the process-only and network-only TON-IoT comparisons in Tables 5.7 and 5.8 provide restricted cross-dataset context, not a controlled test of generalisation across equivalent datasets and feature pipelines. For Objective 3, the held-out classification in Table 5.3 and modality comparison in Table 5.5 support STRIDE-aligned intrusion detection on the evaluated split and telemetry configurations, but only four of six STRIDE categories were evaluated. For Objective 4, Tables 5.10–5.14 support measured residual-evasion reduction for the unified CSA framework under the tested bounded perturbations and oblivious/adaptive gradient-free black-

box models, including TON-IoT host and network data; they do not establish robustness to untested attacks, devices, or environments. The evidence comprises documented partitions and point estimates, not repeated-run stability or statistical significance.

6.13 Discussion Summary

The empirical evidence reviewed in this chapter supports a consistent account of where the framework's performance advantages originate. The strong classification results of Random Forest within the CSA pipeline are not a consequence of classifier selection; they follow from the hybrid Sys-Net profiling layer, which constructs a feature space discriminative enough to resolve cross-domain threats that single-modality systems cannot observe at all. Aligning detection outputs with STRIDE threat semantics strengthens operational interpretability, and the physical consistency verification layer yields adversarial resilience that does not depend on how the classifier was trained (Vassilev et al., 2025; Goodfellow, Shlens, and Szegedy, 2015).

The chapter also examined the wider implications of the work alongside a candid account of its limitations: the Linux-only testbed, the four-category STRIDE scope, and the deliberate restriction to the detection problem. That critical self-assessment locates the contribution accurately within the existing literature and provides the grounding on which the future research directions in Section 6.17 are built.

6.14 Concluding Perspective

Over the past decade, the residential environment has been drawn into a category of infrastructure it was never designed to occupy. Sensors, cameras, thermostats, voice assistants, and intelligent appliances now share a network with each other and with cloud services whose logic governs their behaviour, creating a distributed cyber-physical

system that persists at home and responds continuously to input from residents and remote servers alike (Alrawi et al., 2019). The security implications of this shift are qualitatively different from those of conventional IT deployments. Classical enterprise threat models rest on assumptions, a well-defined perimeter, a shared operating platform, direct administrative control, that characterise none of the environments examined in this thesis (Alrawi et al., 2019).

The pages that follow draw together the principal outcomes of the work. They trace how each research objective was addressed, describe the contributions made to knowledge and to engineering practice, acknowledge the conditions under which those contributions hold, and identify the directions that the infrastructure established here makes tractable for future research.

6.15 Synthesis of Research and Fulfilment of Objectives

The central research aim was to design, implement, and empirically validate a Continuous Security Assurance framework capable of delivering real-time, cross-layer threat intelligence in smart-home IoT environments. Realising that aim required working through a programme that spanned dataset construction, framework design, and controlled experimental evaluation across four research objectives.

Objective 1 required a hybrid profiling architecture merging host-level system telemetry with network-layer traffic statistics into a single, time-synchronised feature representation. The ablation experiments in Chapter 5 establish the necessity of that combination directly. Models based only on host telemetry reported lower performance on Spoofing, while network-only models showed degraded performance for Tampering and Information Disclosure. Across all four evaluated attack categories, the synchronised hybrid configuration achieved the highest F1-score, confirming that Sys-Net fusion provides

a consistent advantage for broad-spectrum security assurance rather than an incremental refinement.

Closely related to the profiling objective was the construction of SmartHome-StrideDS. At the time of assembly, no publicly available IoT security corpus combined time-synchronised, cross-layer telemetry with STRIDE-aligned ground-truth labels (Ring et al., 2019). The dataset was collected from physical smart-home hardware under both benign and adversarial conditions, with each attack episode designed to suppress detectable artefacts in at least one of the two telemetry planes, directly exercising the cross-layer detection hypothesis the framework was built to test.

Objective 2 concerned the detection gain attributable to profiling-grounded anomaly assessment. The comparison between hybrid and single-modality configurations established that the improvement in threat observability originates in the cross-layer integration of features rather than in the choice of classifier. That attribution is important because it means the advantage of the hybrid approach is architectural and principled rather than an artefact of experimental design or algorithm selection. A classification accuracy gain that survives the substitution of any of several different classifiers is evidence of a good feature space, not merely a well-tuned model.

Objective 3 required contextualising detected anomalies within the STRIDE framework. An overall F1-score of 0.96, with per-class F1-scores consistently at or above 0.89 across all four categories, confirms that feature-level deviations can be reliably mapped to specific threat categories. The IDS layer produces alerts carrying semantic content, Denial of Service, Spoofing, Tampering, Information Disclosure, rather than undifferentiated alarm signals, which constitutes a meaningful operational advance toward the adaptive, assurance-grounded detection the CSA model requires.

Objective 4 addressed adversarial robustness. Evaluated against both oblivious and adaptive attacker models on three independent datasets, the physical consistency verification layer substantially reduced net attack success rates while retaining

the same verification architecture and re-deriving domain-appropriate causal rules for the external corpora. The cross-dataset results therefore support the portability of the physical-consistency verification principle across the evaluated feature spaces, subject to corpus-appropriate rule calibration (Vassilev et al., 2025; Goodfellow, Shlens, and Szegedy, 2015).

The optimization of the monitoring intervals for maximum detection responsiveness while being able to handle the computational load was not studied in the primary experimental programme. SmartHome-StrideDS offers such an investigation as a viable trajectory with its time synchronised architecture, though systematic studies of adaptive sampling strategies are still future areas of investigation. Within the scope of the present study the following four research objectives mentioned in Chapter 1 were being realised.

6.16 Key Contributions to Knowledge and Practice

The contributions of this thesis operate at conceptual, empirical, and engineering levels, and it is worth distinguishing them because they make different kinds of claims.

Conceptually, the most significant contribution is the formalisation of the Sys-Net Blind Spot as an analytically distinct structural limitation of single-modality IoT security monitoring. Naming and characterising this gap provides a diagnostic lens through which the adequacy of any proposed security architecture can be assessed: does it model the relationship between system-level and network-level behaviour, or does it treat each plane as a self-contained analytical object? The CSA framework's answer is that cross-layer correlation is not optional; that claim is not asserted but demonstrated through controlled ablation under four STRIDE threat categories.

The CSA framework itself is a second conceptual contribution. It departs from the episodic, event-triggered logic of classical intrusion detection in favour of a continuously quantified security posture measure, one that asks not whether a particular event

is anomalous but whether the system as a whole is operating within its expected security envelope. This reconceptualisation has practical consequences: it transforms security assessment from a reactive response to identified incidents into a proactive characterisation of operational state that is available to operators at all times, not only when an alert fires.

The empirical contribution is SmartHome-StrideDS. Its value as a community resource lies not only in the data it contains but in the labelling scheme it applies: STRIDE-aligned ground-truth annotations combined with time-synchronised dual-plane telemetry enable comparative evaluations that no existing public corpus could previously support (Ring et al., 2019).

A further engineering contribution is the physical consistency verification layer underpinning the framework's adversarial robustness. Rather than learning to resist adversarial perturbations from data, this layer encodes physical and causal constraints that any real-world attack must satisfy regardless of how its feature-space representation is manipulated. The consequence is a form of robustness whose portability was evaluated across datasets with corpus-appropriate rule calibration: tested against SmartHome-StrideDS and two independent TON-IoT corpora under both oblivious and adaptive attacker models, the layer substantially reduced net attack success rates throughout (Vassilev et al., 2025; Goodfellow, Shlens, and Szegedy, 2015).

Finally, the CSA monitoring dashboard addresses the gap between machine-generated threat intelligence and the situational awareness needs of non-specialist operators. Security posture is rendered as a live, interpretable display that conveys both the current classification of device behaviour and its trend over time, without requiring the operator to interpret raw telemetry or statistical anomaly scores directly.

6.17 Future Research Directions

Throughout this thesis, the CSA framework was deliberately bounded at the detection and assurance layer. Automated response and incident handling were withheld on principled grounds: reliable detection is a necessary precondition for any defensible response, and demonstrating it in a heterogeneous Sys-Net environment is itself a non-trivial scientific undertaking. With detection demonstrated for four STRIDE categories under both standard and adversarial conditions, several directions now constitute natural next steps.

Completing STRIDE coverage is the most direct extension. Repudiation detection requires augmenting the telemetry pipeline with structured audit logging via `auditd` or eBPF-based system-call tracing. Elevation of Privilege detection requires supplementing the existing feature set with privilege and capability metadata. Closing this gap would align the framework's empirical coverage with the full threat taxonomy it nominally targets.

Federated deployment addresses the data-governance problem that prevents centralised, cross-household collection of Sys-Net telemetry at scale. In a federated arrangement, each home trains locally on its own device population and shares model updates, rather than raw data, with a shared global model. Per-device baselines (μ_j, σ_j) would thereby reflect the genuine variability of multiple residential environments rather than the single installation from which SmartHome-StrideDS was collected.

Adaptive monitoring intervals represent a practically significant optimisation. A sampling scheme in which polling frequency rises as the anomaly score $A(t)$ approaches the detection threshold θ, and relaxes when $\mathrm{CSA}(t)$ remains stable, would concentrate measurement effort during threat episodes while reducing steady-state computational load. Paired with automatic threshold tightening during periods of elevated posture risk, such a scheme would reduce false-negative rates precisely when they are most operationally consequential.

The adversarial evaluation warrants two extensions. Gradient-based white-box evaluation would establish a ceiling on adversarial exposure under the strongest plausible threat model. Data-poisoning evaluation would characterise the framework's vulnerability to a qualitatively different attack class, one targeting the profiling phase rather than inference, requiring distinct defensive strategies commensurate with that threat.

Explainability is a further practical priority. Incorporating SHAP values into the RF_{IDS} output would expose, for each detection event, the per-feature contribution to the classification decision. This would be particularly valuable for distinguishing between devices flagged through superficially similar feature combinations arising from qualitatively different underlying attack vectors, and would strengthen the framework's utility for forensic analysis alongside its existing role in real-time monitoring.

Automated response is the most consequential architectural extension. The modular design of the CSA pipeline, in which the Phase 4 posture score is already decoupled from the detection decision, was expressly intended to accommodate a response layer without requiring modification to the detection logic that precedes it. Candidate interventions include network isolation of devices whose posture score falls below a defined threshold, suspension of processes flagged through host-level telemetry, and structured operator notification carrying STRIDE-labelled threat context. With the detection foundation in place, this extension becomes an engineering task rather than a research one.

6.18 Concluding Remarks

The inadequacy of single-plane IoT security monitoring is not a matter of implementation quality or parameter choice. It is a consequence of treating network-layer observation and host-level analysis as independent activities, each sufficient on its own, when the nature of modern residential IoT attacks requires that they be examined together. Treating network-layer observation and host-level analysis as independent, self-contained activities

produces a class of threats that neither can see. The CSA framework demonstrates, through controlled experiment rather than assertion, that cross-layer correlation closes this gap in a way that is measurable, reproducible, and robust to deliberate evasion.

The adversarial robustness evaluation adds a dimension that is rarely achieved in IoT security research. Physical consistency verification, which encodes physical law rather than learned statistical regularities, sustains detection coverage against evasive attackers even in scenarios where the underlying classifier has been successfully deceived. No manipulation of feature values in a loss function can produce a sample that violates the physical and causal constraints governing real IoT device behaviour, and those constraints are precisely what the verification layer encodes.

The frameworks, datasets, and empirical results assembled in this thesis are offered as a principled foundation for the next generation of smart-home security tooling. As IoT deployments grow in scale and heterogeneity, and as adversarial techniques targeting residential infrastructure continue to mature, the capacity for continuous, cross-layer, semantically grounded security assurance will cease to be a research aspiration and become an engineering necessity.

References

Abad, Cristina L. and Rafael I. Bonilla (2007). "An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks". In: *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*. IEEE, pp. 60–60. DOI: 10.1109/ICDCSW.2007.19.

Acar, Abbas et al. (2020). "Peek-a-Boo: I See Your Smart Home Activities, Even Encrypted!" In: *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. Association for Computing Machinery, pp. 207–218. DOI: 10.1145/3395351.3399421. URL: https://dl.acm.org/doi/10.1145/3395351.3399421.

Adhikari, R., M. A. Khan, and M. M. Hassan (2024). "A Comprehensive Survey on Anomaly Detection Techniques for IoT Systems". In: *IEEE Internet of Things Journal*.

Ahmad, Zeeshan et al. (2021). "Network intrusion detection system: A systematic study of machine learning and deep learning approaches". In: *Transactions on Emerging Telecommunications Technologies* 32.1, e4150. DOI: 10.1002/ett.4150.

Alajaji, Abdulaziz et al. (2026). "Feature-Space Adversarial Robustness Evaluation of Lightweight ML-Based Intrusion Detection for IoT and Healthcare-Sensing Networks". In: *IEEE Access* 14, pp. 141428–141454. DOI: 10.1109/ACCESS.2026.3732098.

Alars, Estabraq Saleem Abduljabbar and Sefer Kurnaz (2024). "Enhancing Network Intrusion Detection Systems with Combined Network and Host Traffic Features Using Deep Learning: Deep Learning and IoT Perspective". In: *Discover Computing* 27, p. 39. DOI: 10.1007/s10791-024-09480-3.

Alfahaid, Abdullah et al. (2025). "Machine Learning-Based Security Solutions for IoT Networks: A Comprehensive Survey". In: *Sensors* 25.11, p. 3341. DOI: 10.3390/s25113341.

Alhajjar, Elie, Paul Maxwell, and Nathaniel D. Bastian (2021). "Adversarial Machine Learning in Network Intrusion Detection Systems". In: *Expert Systems with Applications* 186, p. 115782. DOI: 10.1016/j.eswa.2021.115782. URL: https://www.sciencedirect.com/science/article/pii/S0957417421011507.

Alrawi, Omar et al. (2019). "SoK: Security Evaluation of Home-Based IoT Deployments". In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 1362–1380. DOI: 10.1109/SP.2019.00013.

AlSalem, Thanaa Saad, Mohammed Amin Almaiah, and Abdalwali Lutfi (2023). "Cybersecurity Risk Analysis in the IoT: A Systematic Review". In: *Electronics* 12.18, p. 3958. DOI: 10.3390/electronics12183958.

Andresini, Giuseppina et al. (2021). "INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection". In: *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*. ACM, pp. 111–122. DOI: 10.1145/3474369.3486864.

Angermeir, Florian et al. (2024). "Towards Automated Continuous Security Compliance". In: *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 440–446. DOI: 10.1145/3674805.3690748. URL: https://dl.acm.org/doi/10.1145/3674805.3690748.

Anthi, Eirini et al. (2019). "A Supervised Intrusion Detection System for Smart Home IoT Devices". In: *IEEE Internet of Things Journal* 6.5, pp. 9042–9053. DOI: 10.1109/JIOT.2019.2926365.

Aparcana-Tasayco, Andres J., Xianjun Deng, and Jong Hyuk Park (2025). "A Systematic Review of Anomaly Detection in IoT Security: Towards Quantum Machine Learning Approach". In: *EPJ Quantum Technology* 12.1, p. 112. DOI: 10.1140/epjqt/s40507-025-00414-6. URL: https://link.springer.com/article/10.1140/epjqt/s40507-025-00414-6.

Apruzzese, Giovanni et al. (2022). "Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems". In: *Digital Threats: Research and Practice* 3.3, pp. 1–19. DOI: 10.1145/3469659.

Araya, Juan Ignacio Iturbe and Helena Rifà-Pous (2023). "Anomaly-based cyberattacks detection for smart homes: A systematic literature review". In: *Internet of Things* 22, p. 100792. DOI: 10.1016/j.iot.2023.100792.

Ardagna, Claudio A. et al. (2017). "A Case for IoT Security Assurance". In: *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*. Springer Singapore, pp. 175–192. DOI: 10.1007/978-981-10-5861-5_8.

Asonze, Christopher Uzoma et al. (2024). "Evaluating the Trade-Offs between Wireless Security and Performance in IoT Networks: A Case Study of Web Applications in AI-Driven Home Appliances". In: *Journal of Engineering Research and Reports* 26.8, pp. 411–432. DOI: 10.9734/jerr/2024/v26i81255. URL: https://journaljerr.com/index.php/JERR/article/view/1255.

Awad, Zeinab, Magdy Zakaria, and Rasha Hassan (2025). "An Enhanced Ensemble Defense Framework for Boosting Adversarial Robustness of Intrusion Detection Systems". In: *Scientific Reports* 15, p. 14177. DOI: 10.1038/s41598-025-94023-z.

Bakhshi, Taimur, Bogdan Ghita, and Ievgeniia Kuzminykh (2024). "A Review of IoT Firmware Vulnerabilities and Auditing Techniques". In: *Sensors* 24.2, p. 708. DOI: 10.3390/s24020708.

Ben Slimane, Jihane, Eman H. Abd-Elkawy, and Albia Maqbool (2024). "Intrusion detection using network traffic profiling and machine learning for IoT". In: *Journal of Electrical Systems* 20.3s, pp. 2140–2149. DOI: 10.52783/jes.1813.

Biggio, Battista and Fabio Roli (2018). "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning". In: *Pattern Recognition* 84, pp. 317–331. DOI: 10.1016/j.patcog.2018.07.023.

Bitdefender and NETGEAR (Oct. 2025). *The Bitdefender and NETGEAR 2025 IoT Security Landscape Report*. Industry Threat Intelligence Report. URL: https://www.bitdefender.com/en-us/news/new-bitdefender-and-netgear-report-reveals-rising-threats-across-the-connected-home-landscape (Accessed: 24 September 2026).

Botto-Tobar, Miguel, Sumaiya Rehan, and Ravi Prakash Verma (2023). "Protecting Smart Home from Cybersecurity Threats Strategies for Homeowners". In: *Journal of Cybersecurity and Information Management* 12.2, pp. 83–98. DOI: 10.54216/JCIM.120206.

Breiman, Leo (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324.

Bridges, Robert A. et al. (2020). "A Survey of Intrusion Detection Systems Leveraging Host Data". In: *ACM Computing Surveys* 52.6, pp. 1–35. DOI: 10.1145/3344382.

Buczak, Anna L. and Erhan Guven (2016). "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection". In: *IEEE Communications Surveys & Tutorials* 18.2, pp. 1153–1176. DOI: 10.1109/COMST.2015.2494502.

Buil-Gil, David et al. (2023). "The digital harms of smart home devices: A systematic literature review". In: *Computers in Human Behavior* 145, p. 107770. DOI: 10.1016/j.chb.2023.107770.

Cambiaso, Enrico et al. (2013). "Slow DoS attacks: definition and categorisation". In: *International Journal of Trust Management in Computing and Communications* 1.3/4, p. 300. DOI: 10.1504/IJTMCC.2013.056440.

Canavese, Daniele et al. (2024). "Security at the Edge for Resource-Limited IoT Devices". In: *Sensors* 24.2, p. 590. DOI: 10.3390/s24020590.

Carlini, Nicholas and David Wagner (2017). "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, pp. 3–14. DOI: 10.1145/3128572.3140444.

Chaabouni, Nadia et al. (2019). "Network Intrusion Detection for IoT Security Based on Learning Techniques". In: *IEEE Communications Surveys & Tutorials* 21.3, pp. 2671–2701. DOI: 10.1109/COMST.2019.2896380.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar (2009). "Anomaly detection: A survey". In: *ACM Computing Surveys* 41.3, pp. 1–58. DOI: 10.1145/1541880.1541882.

Chen, Zhiyan et al. (2023). "Host-Based Network Intrusion Detection via Feature Flattening and Two-stage Collaborative Classifier". In: *arXiv preprint arXiv:2306.09451*. DOI: 10.48550/arXiv.2306.09451. URL: https://arxiv.org/abs/2306.09451.

Chui, Michael, Mark Collins, and Mark Patel (Nov. 2021). *IoT value set to accelerate through 2030: Where and how to capture it*. McKinsey Digital. URL: https://www.mckinsey.com/capabilities/tech-and-ai/our-insights/iot-value-set-to-accelerate-through-2030-where-and-how-to-capture-it (Accessed: 17 August 2026).

Diana, L., P. Dini, and D. Paolini (2025). "Overview on Intrusion Detection Systems for Computers Networking Security". In: *Computers* 14.3, p. 87. DOI: 10.3390/computers14030087.

Díez-Franco, Irene, Pablo García Bringas, and Xabier Ugarte-Pedrero (2024). "Understanding the Security Landscape of Control-Data and Non-Control-Data Attacks Against IoT Systems". In: *2024 9th International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE, pp. 01–06. DOI: 10.23919/SpliTech61897.2024.10612517.

Dinesh, Gayathri et al. (2026). "DA-DCC-IDS: Trustworthy Open-Set IoT Intrusion Detection under Temporal and Device Uncertainty". In: *IEEE Access* 14, pp. 150978–151004. DOI: 10.1109/ACCESS.2026.3732976.

Dobrilovic, Dalibor and Rade Dragović (2024). "Usage of Threat Modelling Tools for Implementation of ICT Security Standards in IoT Environments". In: *International Workshop on Information, Computation, and Control Systems for Distributed Environments*, pp. 45–52. DOI: 10.47350/ICCS-DE.2024.06. URL: https://doi.org/10.47350/ICCS-DE.2024.06.

Engelen, Gints, Vera Rimmer, and Wouter Joosen (2021). "Troubleshooting an Intrusion Detection Dataset: the CICIDS2017 Case Study". In: *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, pp. 7–12. DOI: 10.1109/SPW53761.2021.00009.

Farhan, M. et al. (2025). "Network-based intrusion detection using deep learning technique". In: *Scientific Reports* 15.1, p. 25550. DOI: 10.1038/s41598-025-08770-0.

Federal Bureau of Investigation (June 2025). *Home Internet Connected Devices Facilitate Criminal Activity*. Public Service Announcement, Alert Number I-060525-PSA. URL: https://www.fbi.gov/investigate/cyber/alerts/2025/home-internet-connected-devices-facilitate-criminal-activity (Accessed: 22 September 2026).

Federal Communications Commission (Mar. 2024). *FCC Adopts Rules for IoT Cybersecurity Labeling Program*. URL: https://www.fcc.gov/document/fcc-adopts-rules-iot-cybersecurity-labeling-program (Accessed: 24 September 2026).

Ferrag, Mohamed A., Leandros Maglaras, and Holger Janicke (2023). "Explainable Machine Learning for Smart Home Security: Random Forest and Deep Learning Approaches". In: *IEEE Access* 11, pp. 34567–34581.

Ferrag, Mohamed Amine et al. (2020). "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study". In: *Journal of Information Security and Applications* 50, p. 102419. DOI: 10.1016/j.jisa.2019.102419.

Al-Fuqaha, Ala et al. (2015). "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". In: *IEEE Communications Surveys & Tutorials* 17.4, pp. 2347–2376. DOI: 10.1109/COMST.2015.2444095.

Gama, João et al. (2014). "A survey on concept drift adaptation". In: *ACM Computing Surveys* 46.4, pp. 1–37. DOI: 10.1145/2523813.

Gavric, Nikola, Guru Prasad Bhandari, and Andrii Shalaginov (2026). "Towards Resource-Efficient DDoS Detection in IoT: Leveraging Feature Engineering of System and Network Usage Metrics". In: *Journal of Network and Systems Management* 32.4, p. 69. DOI: 10.1007/s10922-024-09848-2.

Géron, Aurélien (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 3rd ed. O'Reilly Media. ISBN: 9781098125974. URL: https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations (ICLR)*.

Gould, Tara (May 2025). *PumaBot: Novel Botnet Targeting IoT Surveillance Devices*. URL: https://www.darktrace.com/blog/pumabot-novel-botnet-targeting-iot-surveillance-devices (Accessed: 22 September 2026).

Grand View Research (Oct. 2024a). *IoT Security Market (2025 - 2030): Size, Share & Trends Analysis Report*. https://www.grandviewresearch.com/industry-analysis/internet-of-things-iot-security-market. (Accessed: 23 September 2026).

— (2024b). *Smart Home Market Size And Share, Industry Report, 2030*. https://www.grandviewresearch.com/industry-analysis/smart-homes-industry. (Accessed: 5 January 2026).

Al-Hawawreh, Maha et al. (2024). "Evaluating Machine Learning Methods for Smart Home Intrusion Detection: Explainability and Efficiency". In: *Journal of Network and Computer Applications* 230, p. 105635.

He, Haibo and Edwardo A. Garcia (2009). "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9, pp. 1263–1284. DOI: 10.1109/TKDE.2008.239.

Hussain, Fatima et al. (2020). "Machine Learning in IoT Security: Current Solutions and Future Challenges". In: *IEEE Communications Surveys & Tutorials* 22.3, pp. 1686–1721. DOI: 10.1109/COMST.2020.2986444.

Hussain, Muhammad Zunnurain et al. (2026). "LIGHTSEC: Lightweight Explainable IDS for LLNs Using SHAP-RFE-BPSO and Hybrid Ensemble Learnings". In: *IEEE Internet of Things Journal*. Early Access. DOI: 10.1109/JIOT.2026.3732542.

Hwang, Ren-Hung et al. (2023). "Host-based intrusion detection with multi-datasource and deep learning". In: *Journal of Information Security and Applications* 78, p. 103625. DOI: 10.1016/j.jisa.2023.103625.

Imran, Shahbaz Ali et al. (2026). "From Data to Decision: A Framework for Continuous Security Assurance and Live Security Posture Management in IoT". In: *Advances in Intelligent Computing Techniques and Applications II*. Vol. 292. Lecture Notes on Data Engineering and Communications Technologies. Springer Nature Switzerland, pp. 20–30. DOI: 10.1007/978-3-032-23021-8_2.

IoT Security Foundation (2021). *IoT Security Assurance Framework Release 3.0*. https://www.iotsecurityfoundation.org/wp-content/uploads/2021/11/IoTSF-IoT-Security-Assurance-Framework-Release-3.0-Nov-2021-1.pdf. (Accessed: 6 October 2024).

Jawad, Md. Asef et al. (2026). "BRIGHT: Broker-Less Residential IoT Gateway Over HTTPS for Transmission". In: *IEEE Access* 14, pp. 138026–138041. DOI: 10.1109/ACCESS.2026.3730690.

Jeyasoundari, Jeyaraman et al. (2026). "Edge-based man-in-the-middle attack detection in advanced metering infrastructure using a hybrid CNN–GRU–LSTM model". In: *Sādhanā* 51.3, p. 218. DOI: 10.1007/s12046-026-03230-5.

Job, Deepu and Varghese Paul (2021). "Challenges, Security Mechanisms, and Research Areas in IoT and IioT". In: *Internet of Things and Its Applications*. Ed. by Sachi Nandan Mohanty, Jyotir Moy Chatterjee, and Suneeta Satpathy. EAI/Springer Innovations in Communication and Computing. Springer, Cham, pp. 523–538. DOI: 10.1007/978-3-030-77528-5_28.

Kabir, Sohag, Prosanta Gope, and Saraju P. Mohanty (2023). "A security-enabled safety assurance framework for IoT-based smart homes". In: *IEEE Transactions on Industry Applications* 59.1, pp. 6–14. DOI: 10.1109/TIA.2022.3176257.

Kantchelian, Alex, J. D. Tygar, and Anthony D. Joseph (2016). "Evasion and Hardening of Tree Ensemble Classifiers". In: *Proceedings of the 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. PMLR, pp. 2387–2396. URL: https://proceedings.mlr.press/v48/kantchelian16.html.

Khaleghi, Bahador et al. (2013). "Multisensor Data Fusion: A Review of the State-of-the-Art". In: *Information Fusion* 14.1, pp. 28–44. DOI: 10.1016/j.inffus.2011.08.001.

Khraisat, Ansam and Ammar Alazab (2021). "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges". In: *Cybersecurity* 4.1, p. 18. DOI: 10.1186/s42400-021-00077-7.

Khraisat, Ansam, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman (2019). "Survey of intrusion detection systems: techniques, datasets and challenges". In: *Cybersecurity* 2.1, p. 20. DOI: 10.1186/s42400-019-0038-7.

Khraisat, Ansam, Iqbal Gondal, Peter Vamplew, Joarder Kamruzzaman, and Ammar Alazab (2019). "A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks". In: *Electronics* 8.11, p. 1210. DOI: 10.3390/electronics8111210.

Kikissagbe, B. R. and Meddi Adda (2024). "Machine learning-based intrusion detection methods in IoT systems: A comprehensive review". In: *Electronics* 13.18, p. 3601. DOI: 10.3390/electronics13183601.

Koroniotis, Nickolaos et al. (2019). "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset". In: *Future Generation Computer Systems* 100, pp. 779–796. DOI: 10.1016/j.future.2019.05.041.

Kumar, Vivek, Kamal Kumar, and Maheep Singh (2025). "NIDS-CBAD: Detecting Adversarial Attacks in Network Intrusion Detection Systems Using Domain Constraints". In: *International Journal of Machine Learning and Cybernetics* 16, pp. 4213–4234. DOI: 10.1007/s13042-024-02506-8.

Lara, Agustín et al. (2023). "Smart home anomaly-based IDS: Architecture proposal and case study". In: *Internet of Things* 22, p. 100773. DOI: 10.1016/j.iot.2023.100773.

Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou (2008). "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

Lo, Carol et al. (2026). "LOTL-hunter: Detecting multi-stage living-off-the-land attacks in cyber-physical systems using decision fusion techniques with digital twins". In: *Future Generation Computer Systems* 180, p. 108382. DOI: 10.1016/j.future.2026.108382.

Madry, Aleksander et al. (2017). "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations (ICLR)*. DOI: 10.48550/arXiv.1706.06083.

Meidan, Yair et al. (2023). "CADeSH: Collaborative Anomaly Detection for Smart Homes". In: *IEEE Internet of Things Journal* 10.10, pp. 8514–8532. DOI: 10.1109/JIOT.2022.3194813.

MITRE (2026). *MITRE ATT&CK*. MITRE. URL: https://attack.mitre.org/ (Accessed: 17 August 2026).

Moustafa, Nour (2021). "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets". In: *Sustainable Cities and Society* 72, p. 102994. DOI: 10.1016/j.scs.2021.102994.

Moustafa, Nour, Mohiuddin Ahmed, and Sherif Ahmed (2020). "Data Analytics-enabled Intrusion Detection: Evaluations of ToN IoT Linux Datasets". In: *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, pp. 727–735. DOI: 10.1109/TrustCom50675.2020.00100.

Mudgerikar, Anand, Puneet Sharma, and Elisa Bertino (2019). "E-Spion: A System-Level Intrusion Detection System for IoT Devices". In: *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*. ACM, pp. 493–500. DOI: 10.1145/3321705.3329857.

Nadifi, Zineb et al. (2025). "STRIDE-Based Threat Modeling and Risk Assessment Framework for IoT-enabled Smart Healthcare Systems". In: *International Journal of Online and Biomedical Engineering (iJOE)* 21.9, pp. 63–80. DOI: 10.3991/ijoe.v21i09.55517.

Narayanan, Abhilash and M. Vanitha (2025). "Apply the Internet of Things for Banking Cybersecurity: Utilize Real-Time Threat Detection and Blockchain Technology to Protect Interconnected Financial Ecosystems". In: *2025 10th International Conference on Smart Structures and Systems (ICSSS)*. IEEE, pp. 1–6. DOI: 10.1109/ICSSS66939.2025.11346420.

National Institute of Standards and Technology (2025). *CVE-2025-1316 Detail*. National Vulnerability Database. URL: https://nvd.nist.gov/vuln/detail/CVE-2025-1316 (Accessed: 24 September 2026).

Neshenko, Nataliia et al. (2019). "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations". In: *IEEE Communications Surveys & Tutorials* 21.3, pp. 2702–2733. DOI: 10.1109/COMST.2019.2910750.

Neupane, Subash et al. (2022). "Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities". In: *IEEE Access* 10, pp. 112392–112415. DOI: 10.1109/ACCESS.2022.3216617.

Ouaissa, Mariya and Mariyam Ouaissa (2025). "Analyzing and Mitigating Attacks in IoT Smart Home Using a Threat Modeling Approach-Based STRIDE". In: *International Journal of Interactive Mobile Technologies (iJIM)* 19.2, pp. 126–142. DOI: 10.3991/ijim.v19i02.52377.

Pang, Guansong et al. (2021). "Deep Learning for Anomaly Detection: A Review". In: *ACM Computing Surveys* 54.2, pp. 1–38. DOI: 10.1145/3439950. URL: https://dl.acm.org/doi/10.1145/3439950.

Papadogiannaki, Eva and Sotiris Ioannidis (2021). "A Survey on Encrypted Network Traffic Analysis Applications, Techniques, and Countermeasures". In: *ACM Computing Surveys* 54.6. Article 123, pp. 1–35. DOI: 10.1145/3457904.

Papernot, Nicolas et al. (2017). "Practical Black-Box Attacks against Machine Learning". In: *Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security (ASIACCS)*, pp. 506–519. DOI: 10.1145/3052973.3053009.

Pascoe, Cherilyn, Stephen Quinn, and Karen Scarfone (2024). *The NIST Cybersecurity Framework (CSF) 2.0*. Tech. rep. NIST CSWP 29. National Institute of Standards and Technology. DOI: 10.6028/NIST.CSWP.29. URL: https://www.nist.gov/publications/nist-cybersecurity-framework-csf-20 (Accessed: 22 September 2026).

Pendleton, Marcus et al. (2016). "A Survey on Systems Security Metrics". In: *ACM Computing Surveys* 49.4, pp. 1–35. DOI: 10.1145/3005714.

Ring, Markus et al. (2019). "A Survey of Network-Based Intrusion Detection Data Sets". In: *Computers & Security* 86, pp. 147–167. DOI: 10.1016/j.cose.2019.06.005. URL: https://www.sciencedirect.com/science/article/pii/S016740481930118X.

Robertson, Derek and Matthew Kosinski (Apr. 2026). *What Is Configuration Drift?* https://www.ibm.com/think/topics/configuration-drift. IBM Think, 29 April 2026. IBM. (Accessed: 23 September 2026).

Ron, Diego Arguello et al. (2025). "Anomaly Detection Tools for the Lifecycle Security of Smart Systems". In: *2025 IEEE International Conference on Smart Computing (SMARTCOMP)*. Cork, Ireland: IEEE, pp. 276–281. DOI: 10.1109/SMARTCOMP65954.2025.00086.

Rosenberg, Ishai et al. (2021). "Adversarial Machine Learning Attacks and Defense Methods in the Cyber Security Domain". In: *ACM Computing Surveys* 54.5, pp. 1–36. DOI: 10.1145/3453158.

Saleem, Malik et al. (2022). "Machine learning-based intrusion detection systems: A contemporary survey". In: *Journal of Network and Computer Applications*.

Samek, Wojciech, Grégoire Montavon, Sebastian Lapuschkin, et al. (2021). "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications". In: *Proceedings of the IEEE* 109.3, pp. 247–278. DOI: 10.1109/JPROC.2021.3060483. URL: https://ieeexplore.ieee.org/document/9369420.

Samek, Wojciech, Grégoire Montavon, Andrea Vedaldi, et al., eds. (2019). *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Vol. 11700. Lecture Notes in Computer Science. Springer Nature. DOI: 10.1007/978-3-030-28954-6.

Sarwar, Nadeem et al. (2023). "IoT Network Anomaly Detection in Smart Homes Using Machine Learning". In: *IEEE Access* 11, pp. 119462–119480. DOI: 10.1109/ACCESS.2023.3325929.

Shafiq, Muhammad et al. (2021). "CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques". In: *IEEE Internet of Things Journal* 8.5, pp. 3242–3254. DOI: 10.1109/JIOT.2020.3002255.

Sheatsley, Ryan et al. (2021). "On the Robustness of Domain Constraints". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 495–515. DOI: 10.1145/3460120.3484570.

Shevchenko, Nataliya et al. (Aug. 2018). *Threat Modeling: A Summary of Available Methods*. Tech. rep. Software Engineering Institute, Carnegie Mellon University. URL: https://www.sei.cmu.edu/library/threat-modeling-a-summary-of-available-methods/ (Accessed: 17 August 2026).

Shostack, Adam (2014). *Threat Modeling: Designing for Security*. Indianapolis, IN: John Wiley & Sons. ISBN: 978-1-118-80999-0.

Shukla, Ankur et al. (2022). "System Security Assurance: A Systematic Literature Review". In: *Computer Science Review* 45, p. 100496. DOI: 10.1016/j.cosrev.2022.100496.

Sivanathan, Arunan et al. (2019). "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics". In: *IEEE Transactions on Mobile Computing* 18.8, pp. 1745–1759. DOI: 10.1109/TMC.2018.2866249.

Slack, Dylan et al. (2020). "Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. ACM, pp. 180–186. DOI: 10.1145/3375627.3375830.

Smiliotopoulos, Christos et al. (2026). *Bridging Explanation and Operations: A Systematic Review and Taxonomy of XAI-Enabled Cybersecurity Detection*. Research Square preprint (not peer reviewed). DOI: 10.21203/rs.3.rs-10669512/v1.

Softcat (Feb. 2024). *Continuous Threat Exposure Management (CTEM): a comprehensive approach to cyber security*. https://www.softcat.com/blog/continuous-threat-exposure-management-ctem-comprehensive-approach-cyber-security. (Accessed: 6 October 2024).

Sommer, Robin and Vern Paxson (2010). "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection". In: *2010 IEEE Symposium on Security and Privacy*, pp. 305–316. DOI: 10.1109/SP.2010.25.

Sovacool, Benjamin K. and Dylan D. Furszyfer Del Rio (2020). "Smart Home Technologies in Europe: A Critical Review of Concepts, Benefits, Risks and Policies". In: *Renewable and Sustainable Energy Reviews* 120, p. 109663. DOI: 10.1016/j.rser.2019.109663.

Spadafora, Anthony (Oct. 2019). *Smart home devices are being hit with more cyberattacks than ever*. https://www.techradar.com/news/smart-home-devices-are-being-hit-with-more-cyberattacks-than-ever. (Accessed: 4 October 2024).

Stoyanova, Maria et al. (2020). "A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues". In: *IEEE Communications Surveys & Tutorials* 22.2, pp. 1191–1221. DOI: 10.1109/COMST.2019.2962586.

Sun, Panjun et al. (2024). "A Survey on Security Issues in IoT Operating Systems". In: *Journal of Network and Computer Applications* 231, p. 103976. DOI: 10.1016/j.jnca.2024.103976.

Taherdoost, Hamed (2023). "Security and internet of things: benefits, challenges, and future perspectives". In: *Electronics* 12.8, p. 1901. DOI: 10.3390/electronics12081901.

Tait, Melissa (May 2023). *Top 5 Smart Home Security Risks and How to Prevent Them*. https://www.originwirelessai.com/top-5-smart-home-security-risks-and-how-to-prevent-them/. Origin Wireless. (Accessed: 4 October 2024).

Touqeer, Haseeb et al. (2021). "Smart home security: challenges, issues and solutions at different IoT layers". In: *The Journal of Supercomputing* 77.12, pp. 14053–14089. DOI: 10.1007/s11227-021-03825-1.

Tramèr, Florian et al. (2020). "On Adaptive Attacks to Adversarial Example Defenses". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 1633–1645. DOI: 10.48550/arXiv.2002.08347.

Trivedi, Kishor S. (2016). *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. 2nd ed. Hoboken, NJ: John Wiley & Sons. DOI: 10.1002/9781119285441.

Valluru, Viveka Vardhini, Sanyam Shukla, and R. K. Pateriya (2026). "Comparing Multi-Label Approaches for Intrusion Detection: Binary Relevance, Classifier Chains, and

Label Powerset on the ADFA-LD Dataset". In: *2026 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*. IEEE, pp. 1–6. DOI: 10.1109/IATMSI68868.2026.11465681.

Vardakis, George et al. (2024). "Review of Smart-Home Security Using the Internet of Things". In: *Electronics* 13.16, p. 3343. DOI: 10.3390/electronics13163343.

Vassilev, Apostol et al. (2025). *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*. Tech. rep. NIST AI 100-2e2025. National Institute of Standards and Technology. DOI: 10.6028/NIST.AI.100-2e2025. URL: https://csrc.nist.gov/pubs/ai/100/2/e2025/final (Accessed: 22 September 2026).

Verma, Abhishek and Virender Ranga (2020). "Machine Learning Based Intrusion Detection Systems for IoT Applications". In: *Wireless Personal Communications* 111.4, pp. 2287–2310. DOI: 10.1007/s11277-019-06986-8.

Vitorino, João, Nuno Oliveira, and Isabel Praça (2022). "Adaptative Perturbation Patterns: Realistic Adversarial Learning for Robust Intrusion Detection". In: *Future Internet* 14.4, p. 108. DOI: 10.3390/fi14040108.

Watanabe, Naoto et al. (Mar. 2024). "Self-Adaptive Traffic Anomaly Detection System for IoT Smart Home Environments". In: *IEICE Transactions on Communications* E108-B.3, pp. 230–242. DOI: 10.23919/transcom.2024EBT0002.

Winanto, Eko Arip and Mohd Yazid Idris (2026). "Adversarial Robustness of Deep Learning-Based Intrusion Detection Systems: A Systematic Literature Review of Attacks, Defenses, Datasets, and Open Challenges". In: *IEEE Access* 14, pp. 149555–149591. DOI: 10.1109/ACCESS.2026.3733985.

Yaras, Sami and Murat Dener (2024). "IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm". In: *Electronics* 13.6, p. 1053. DOI: 10.3390/electronics13061053.

Yoachimik, Omer and Jorge Pacheco (Jan. 2025). *Record-Breaking 5.6 Tbps DDoS Attack and Global DDoS Trends for 2024 Q4*. URL: https://blog.cloudflare.com/ddos-threat-report-for-2024-q4/ (Accessed: 22 September 2026).

Yuan, Bin et al. (2023). "On the Security of Smart Home Systems: A Survey". In: *Journal of Computer Science and Technology* 38.2, pp. 228–247. DOI: 10.1007/s11390-023-2488-3.

Zarpelão, Bruno Bogaz et al. (2017). "A survey of intrusion detection in Internet of Things". In: *Journal of Network and Computer Applications* 84, pp. 25–37. DOI: 10.1016/j.jnca.2017.02.009.