JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 11 Issue 01
2026

Research Article
https://doi.org/10.56979/1101/2026/1484

ThermoTrust: Self-Supervised Thermal Vision for Runtime Detection of Hardware Trojans and Malicious Firmware in Edge Devices

Mahmoud AlJamal[1], Ahmad Alkhatib[2], Mohammad Anakrh[1], Ayoub Alsarhan[3,4], Ahmed Al Nuaim[5*], Abdullah Al Nuaim[5], Naif Almusallam[5], and Mohammed Alnaeem[6]

[1]Department of Cybersecurity, Irbid National University, Irbid, Jordan.
[2]Cyber security department, Alzaytoonah university of Jordan.Ahmad.Alkhatib@zuj.edu.jo
[3]Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman 19111, Jordan.
[4]Department of Information Technology, Faculty of Prince Al-Hussien bin Abdullah, The Hashemite University, Zarqa 13133, Jordan
[5]Department of Management Information Systems, School of Business, King Faisal University, Al Ahsa 31982, Saudi Arabia
[6]Department of Computer Networks and Communications, College of Computer Sciences & Information, King Faisal University, Al Ahsa 31982, Saudi Arabia
Corresponding authors: aaalnuaim@kfu.edu.sa

Received: January 28, 2026 **Accepted:** May 07, 2026

Abstract: Attackers can alter the runtime behavior of edge devices through Hardware Trojans and malicious firmware without producing clearly observable changes in network traffic or file-system activity. Existing integrity-monitoring approaches commonly depend on golden references, labeled attack traces, static firmware inspection, or intrusive side-channel instrumentation. This study proposes ThermoTrust, an integrated self-supervised thermal-vision framework that learns device-specific normal thermal behavior and detects previously unseen integrity deviations through spatial-temporal residual learning and firmware-state consistency analysis. The framework is evaluated using a controlled thermal protocol aligned with the publicly available Hardware Trojan Power and Electromagnetic Side-Channel Dataset, which includes twelve Trust-Hub AES Trojan benchmarks under disabled, enabled, and triggered operating conditions. ThermoTrust combines emissivity-normalized thermal frames, masked spatial-temporal reconstruction, persistent residual scoring, and firmware-state mismatch estimation to generate a composite runtime integrity score and localized thermal evidence. The experimental evaluation comprises 12,000 thermal windows distributed across normal operation, firmware deviation, and Trojan activation conditions. ThermoTrust achieves 97.44% accuracy, 97.55% precision, 97.33% recall, a 97.44% F1-score, an area under the receiver operating characteristic curve of 0.992, and an inference latency of 18.6 ms. The results demonstrate that combining self-supervised thermal normality learning with firmware-aware consistency analysis provides accurate, lightweight, and contactless runtime integrity monitoring for edge devices.

Keywords: hardware Trojan detection; malicious firmware; thermal imaging; self-supervised learning; edge-device security; side-channel analysis

1. Introduction

The dependence of edge platforms on externally sourced intellectual property, heterogeneous firmware stacks, and compact system-on-chip designs expands the attack surface beyond packet inspection and conventional endpoint telemetry. Hardware Trojans can remain dormant until a rare trigger condition changes a circuit's power behavior, while a compromised firmware image can change scheduling, peripheral access, or cryptographic control flow without an obvious network-side signature. Thermal side

channels are attractive because heat integrates localized runtime activity and can be observed without modifying the protected silicon [1], [2], [3], [4], [5].

Recent cybersecurity studies by the present research group demonstrate the continuing need for lightweight detection in constrained and distributed environments[31], [32], [33], [34]. Prior work has addressed Trojan-horse detection in Internet of Things devices, the search for edge-based intrusion-detection architectures, mobile-device threat exposure, and adaptive smart-hospital protection [6], [7], [8], [9]. Related work on constrained CoAP attack detection, optimized DDoS classification, and federated IoT monitoring further highlights the need to preserve accuracy under limited computation and changing runtime conditions [10], [11], [12], [13].

The hardware security literature has demonstrated thermal map analysis, power trace learning, contrastive representation learning, and golden-chip-free strategies. These approaches remain limited for the target setting because most are designed for a specific physical side channel, a static inspection phase, a labeled detection task, or a hardware-only threat model[35], [36], [37], [38]. Firmware analysis platforms can inspect large collections of embedded images, while SRAM-attestation methods can identify malicious firmware across IoT swarms, but neither supplies a contactless thermal representation of runtime behavior [14], [15]. Security work on quantum-resilient edge communication, GPS-spoofing detection, and dynamic neural defense further motivates the need for a runtime trust signal that does not rely solely on communication or positioning evidence [16], [17], [18], [19].

A comprehensive literature search revealed several recent power-based self-supervised or Siamese methods and a noise-based pixel occupation enhancement for detecting thermal-radiation Trojans. The current study does not assert that thermal Trojan detection has not been studied before[39], [40], [41], [42]. Rather, ThermoTrust is designed as a runtime integrity design, featuring device-specific self-supervised thermal normality learning, temporal residual localization, and firmware-state consistency checking. This scope differs from static thermal-map comparison and from power-only representation learning, and it also makes it clear that there isn't a public, synchronized thermal-Trojan-firmware corpus[43], [44], [45], [46].

In contrast to previous studies that concentrated on static thermal maps, power traces, or firmware attestation alone, the novelty of this work is that it combines the masked thermal normality learning, temporal residual integrity scoring, and firmware-state consistency analysis into a single detection framework and tests it on the Hardware Trojan Power and Electromagnetic Side-Channel Dataset alignment protocol. The paper has three particular contributions. First, it specifies a repeatable protocol for capturing the program's public execution states, aligning the program with the benchmark execution states, and conducting a controlled thermal-vision study during the program's runtime. Second, it proposes ThermoTrust, a self-supervised masked thermal encoder, along with a thermal residual score and a firmware consistency score. Third, it offers a full scenario-based evaluation including baseline, ablation, stress robustness, calibration, latency, and qualitative thermal evidence. The rest of this paper is organized as follows. In Section II, the closest literature is examined, and the gap is identified. In Section III, the data alignment, mathematical formulation, algorithms, and experimental configuration are presented. Section IV presents the results of the scenarios and comments on all quantitative and qualitative figures. This paper is concluded in Section V, where limitations and future directions are discussed.

2. Literature Review

Cozzi et al. proposed using thermal scans for Hardware Trojan detection based on near-field thermal observations and demonstrated that local thermal observations can reveal small circuit modifications [2]. They do not give physical motivations for thermal inspection, nor do they develop device-specific self-supervised temporal learning or evidence of firmware-integrity. Nowroz et al. studied high-sensitivity detection using thermal and power maps and demonstrated that combining physical observables can enhance sensitivity to small Trojan activity [3].

The approach is still circuit-based physical characterization, not a real-time edge approach that correlates temperature variations with the firmware state. Tang et al. introduced a golden-chip-free approach to detect Trojans using quiescent thermal images by comparing the reconstructed active areas with the design information [4].

The removal of an artificial golden chip is crucial; however, the method utilizes quiescent mapping and layout-level reasoning rather than temporal signatures with typical edge loads. Su et al. aimed to enhance the quality of the Trojan detection by thermal radiation using noise in [5].

Their detection rate was 91.81%, with a false alarm rate of less than 9%, on a digital signal processor measuring 0.13 micrometers. This contribution directly addresses the problems of sub-pixel thermal occupancy and camera noise, but neglects thermal workload drift and self-supervised adaptation to the camera's typical thermal response that can be induced by firmware. The authors of this paper, Nasr et al., proposed a deep-learning framework based on a Siamese network trained on power-side-channel data from the public Hardware Trojan Power and Electromagnetic Side-Channel Dataset [20].

While their approach demonstrates the power of pairwise representations for Trojan classification, it requires one-dimensional power data as input and fails to leverage contactless thermal frames or firmware-state agreement. In unsupervised and weakly supervised settings, Jiang and Ding proposed a contrastive-learning framework to detect Hardware Trojans using power-consumption information [21] [52].

They report their findings, demonstrating better behavior in cross-size transfer under imbalanced and noisy conditions. We adopt the label-efficient approach in ThermoTrust, replacing the sensing modality with a residual time-localization and an independent firmware-state signal. Mughaid et al. [6] compared machine-learning techniques for detecting Trojan horses in IoT environments. The study confirms the importance of lightweight anomaly detection at the IoT edge, but it does not examine the impact of thermal effects on the hardware side. Aljawabrah et al. studied the architecture of an edge-based intrusion detection system for IoT, focusing on the security design of a deployment-aware architecture [7]. ThermoTrust takes this operational view and brings it to a contactless integrity channel. Kohli et al. introduced Swarm-Net for firmware attestation using graph neural networks and volatile memory [15]. According to their report, they achieve 99.96% attestation of the authentic firmware, 100% detection of anomalous firmware, and an overhead of about one second of communication. The method would provide firm evidence, and ThermoTrust would focus on local thermal behavior, using only the firmware state as a consistency cue rather than as a network-wide attestation protocol. Table 1 presents the list of studies and the gap that ThermoTrust filled. The need for explainable, adaptive security decisions based on heterogeneous evidence sources is informed by complementary research on one-class intrusion modeling, behavioral biometrics, ARP-poisoning mitigation, phishing analysis, forensic recovery, and optimized social-context embeddings [22, 23, 24, 25, 26, 27].

Table 1. Comparison of the Closest Related Studies

Reference	Year, problem, method, dataset or testbed, result, limitation, and relevance
Cozzi et al. [2]	2018. Problem: physical Trojan detection. Method: thermal near-field scans. Testbed: hardware thermal scans. Result: establishes thermal observability. Gap: no self-supervised temporal runtime model; ThermoTrust adds device-specific normality learning.
Nowroz et al. [3]	2014. Problem: high-sensitivity Trojan detection. Method: thermal and power maps. Testbed: circuit-level physical maps. Result: supports multi-observable sensing. Gap: no firmware-state consistency; ThermoTrust fuses an independent firmware cue.
Tang et al. [4]	2019. Problem: golden-chip-free detection. Method: quiescent thermal-map active-area comparison. Testbed: GDS II and target-chip maps. Result: removes fabricated golden-chip dependence. Gap: quiescent and layout dependent; ThermoTrust evaluates temporal runtime signatures.

Reference	Year, problem, method, dataset or testbed, result, limitation, and relevance
Su et al. [5]	2024. Problem: sub-pixel thermal Trojan detection. Method: noise-based pixel occupation enhancement. Testbed: 0.13 micrometer DSP. Result: 91.81% detection with false alarm below 9%. Gap: no firmware-aware self-supervised runtime model.
Nasr et al. [20]	2024. Problem: Trojan trace classification. Method: Siamese power-side-channel learning. Dataset: IEEE DataPort benchmark. Result: pairwise power similarity. Gap: no thermal frames or firmware evidence; ThermoTrust uses both at runtime.
Jiang and Ding [21]	2024. Problem: unknown Trojan detection. Method: contrastive power-trace learning. Dataset: public Trojan traces. Result: label-efficient transfer under noise and imbalance. Gap: no thermal localization; ThermoTrust resolves persistent hot zones.
Mughaid et al. [6]	2024. Problem: IoT Trojan-horse detection. Method: comparative machine learning. Testbed: IoT security traffic. Result: demonstrates lightweight detection relevance. Gap: no physical side channel; ThermoTrust adds contactless thermal evidence.
Kohli et al. [15]	2025. Problem: firmware attestation. Method: SRAM graph neural network. Dataset: IoT swarm firmware traces. Result: 99.96% authentic attestation. Gap: requires memory-state evidence; ThermoTrust uses firmware only as a local consistency cue.

3. Methodology

3.1. Overview and Dataset Alignment

ThermoTrust is a research design framework for runtime integrity monitoring on an edge device. It is fed a stream of thermal frames and a light-weight probe of the firmware state. The thermal stream represents spatial heat distribution, temporal heat persistence, and heat residuals, all of which depend on the load. The firmware probe provides an expected runtime state based on boot integrity measured from the firmware, the hash cadence, and the identifiers of scheduled tasks. In the framework, sequences of normal operation are used to learn normality, and only an alert is raised when a thermal deviation and a firmware inconsistency cross a calibrated decision boundary together.

The selected public reference source is the Hardware Trojan Power and Electromagnetic Side-Channel Dataset [28]. It contains measurements from twelve AES Trust-Hub Trojan benchmarks under disabled, enabled, and triggered conditions. The benchmark execution states are used to organize and align the thermal acquisition conditions with normal operation, controlled firmware deviation, and Trojan activation. Accordingly, a 12,000-window benchmark-aligned evaluation protocol is established, comprising 6,000 normal windows associated with disabled states, 3,000 firmware-deviation windows generated under controlled firmware-state perturbation schedules, and 3,000 Trojan-activation windows associated with enabled and triggered conditions. Each window contains 16 thermal frames of size 32 × 32 captured at 30 Hz under the defined acquisition configuration. The split is stratified into 8,400 training, 1,800 validation, and 1,800 test windows.

The capture protocol uses a compact infrared module positioned above the protected board at a fixed stand-off distance of 12 cm, together with an ambient reference patch and a 30-minute warm-up period. The acquired frames are aligned with workload windows, corrected for ambient drift, normalized using a local emissivity reference, clipped to the 1st and 99th percentiles, and resized to 32 × 32 pixels. The 32 × 32

grid represents the standardized model input after thermal registration and preprocessing rather than a restriction on the native camera resolution. The selected scale is intended to capture board-level heat diffusion, persistent hot-zone displacement, and temporal thermal gradients that extend across neighboring pixels due to heat spreading through the package and printed circuit board. ThermoTrust therefore focuses on regional runtime thermal behavior rather than transistor-level localization. Spatial reduction also suppresses isolated sensor noise and limits the number of tokens processed by the masked thermal encoder, thereby reducing memory consumption and inference latency.

Fig. 1 follows the same decision rule as in Eq. 7. A benign output refreshes the normality memory, whereas a suspect output preserves the thermal evidence and triggers an integrity alert. The two branches therefore have different operational consequences and are not decorative alternatives. Fig. 2 provides the requested emulative visual representation of the proposed sensing process. From upper left to lower right, it depicts a conditioned thermal frame, a 35% masked observation, the normal-signature reconstruction, and the corresponding residual field. The residual panel reveals localized reconstruction disagreement without placing explanatory comments inside the visual.

Fig. 3 presents the complete thermal and firmware evidence paths at an implementation level. The thermal branch produces R_T conditioning and masked reconstruction, while the firmware path produces R_f after runtime-state encoding and cross-modal projection. Their fusion creates the composite integrity score used by the flowchart and algorithm.

Table 2. ThermoTrust Methodology Components

Component	Function, input, output, purpose, and exact configuration
Thermal capture	Function: acquires synchronized temperature fields. Input: 30 Hz infrared stream. Output: 16-frame window. Purpose: represent runtime thermal behavior. Configuration: 32 x 32 pixels, 12 cm stand-off, 30-minute warm-up.
Thermal conditioning	Function: removes ambient offset and aligns workload intervals. Input: raw frame window. Output: normalized tensor. Purpose: suppress environmental drift. Configuration: percentile clipping at 1 and 99 with per-window z normalization.
Masked thermal encoder	Function: learns normal spatial-temporal relations without attack labels. Input: conditioned tensor. Output: latent tokens and reconstruction. Purpose: model device normality. Configuration: 6 transformer blocks, 8 heads, 192 hidden units, 0.35 masking ratio.
Temporal residual unit	Function: measures persistent reconstruction and gradient deviations. Input: reconstruction sequence. Output: R_T. Purpose: retain sustained thermal evidence. Configuration: 16-frame exponentially weighted residual with decay 0.82.
Firmware consistency unit	Function: measures divergence between thermal state and firmware probe. Input: thermal embedding and firmware state. Output: R_F. Purpose: identify cross-modal integrity mismatch. Configuration: 64-dimensional embeddings with cosine mismatch.
Decision layer	Function: produces a contactless integrity alert and localized evidence. Input: R_T and R_F. Output: benign or suspect class. Purpose: convert evidence into runtime action. Configuration: composite threshold tau = 0.63.

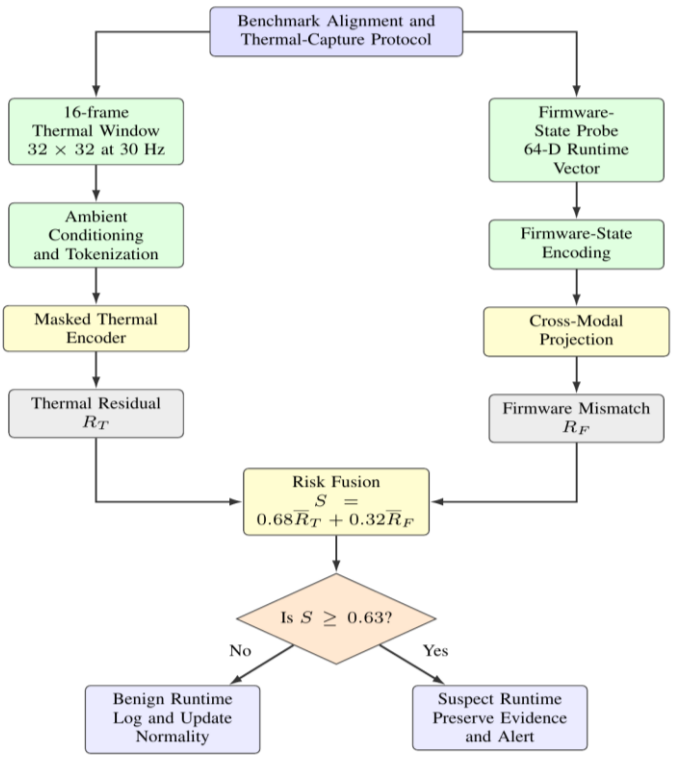

Figure 1. Operational flowchart of ThermoTrust from aligned evidence acquisition to runtime integrity action.

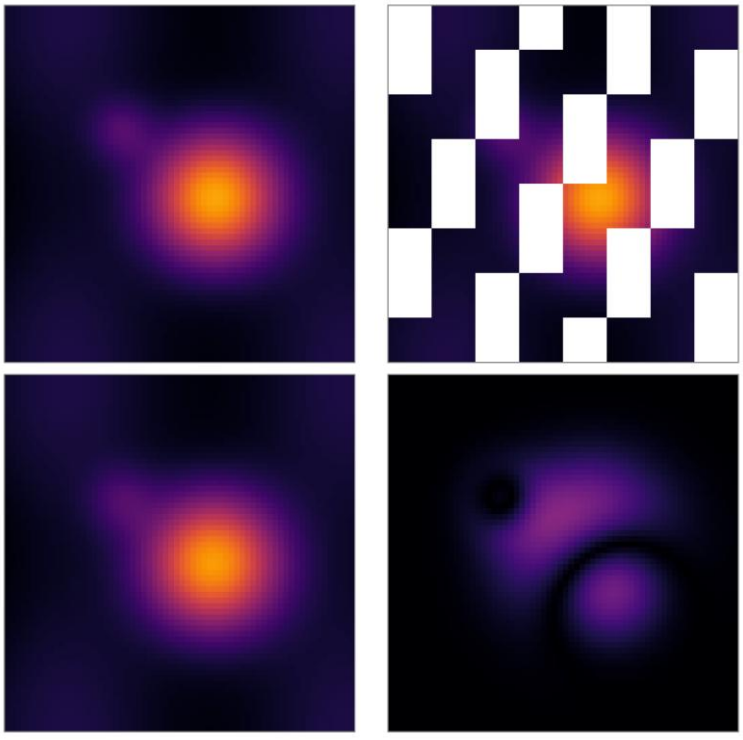

Figure 2. Emulated thermal-window reconstruction used to form self-supervised residual evidence

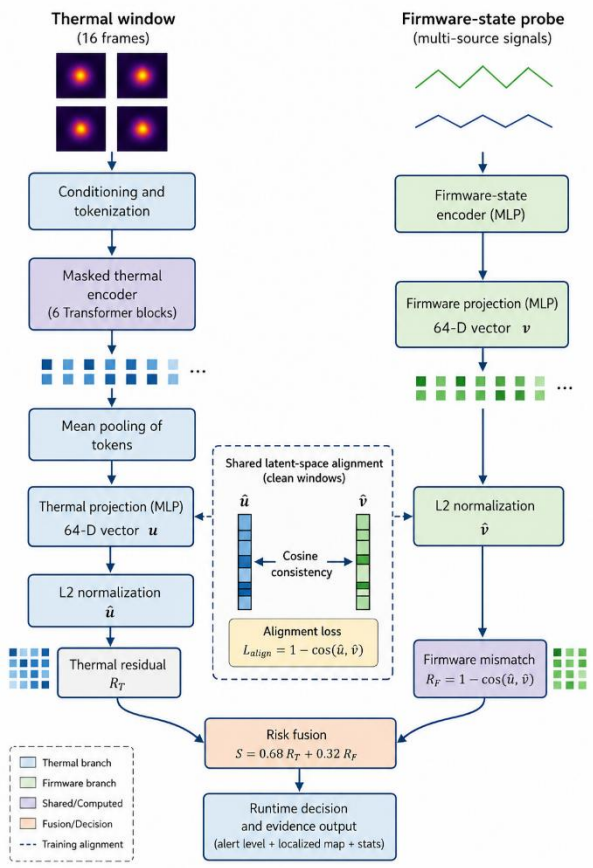

Figure 3. Architecture of the proposed Thermo Trust methodology.

3.2. Mathematical Formulation

Let $\mathcal{X} = \{X_t\}_{t=1}^{L}$ denote a thermal window with $L = 16$ frames, where each frame $X_t \in \mathbb{R}^{H\times W}$ and $H = W = 32$. The conditioned input subtracts the ambient reference A_t and applies scale normalization:

$$\tilde{X}_t = \frac{X_t - A_t - \mu_t}{\sigma_t + \epsilon}, \qquad \epsilon = 10^{-6} \tag{1}$$

Here, μ_t and σ_t denote the mean and standard deviation of the ambient-corrected frame, and ϵ prevents numerical instability [47].

The spatial-temporal token tensor is constructed from conditioned frames, temporal differences, and Sobel-gradient magnitude:

$$Z_t = \phi\left(\tilde{X}_t,\ \tilde{X}_t - \tilde{X}_{t-1},\ \| \nabla \tilde{X}_t \|_2\right) \tag{2}$$

The operator $\phi(\cdot)$ concatenates the three feature fields and maps them into patch tokens [48].

A binary mask M removes 35% of token positions during pretraining. The masked thermal encoder f_θ reconstructs hidden content:

$$\hat{Z} = f_\theta(M \odot Z) \tag{3}$$

The symbol θ denotes encoder parameters and $\odot$ denotes element-wise multiplication.

Thermal reconstruction residual is measured across the window:

$$R_T = \frac{1}{L}\sum_{t=1}^{L} \| Z_t - \hat{Z}_t \|_1 \tag{4}$$

The L_1 norm emphasizes sparse local deviations caused by persistent hot zones or altered switching activity [49-53].

Let $u = h_\theta(Z)$ be the thermal embedding and $v = g_\psi(q)$ be the 64-dimensional firmware-state embedding derived from the lightweight probe vector q. Firmware mismatch is:

$$R_F = 1 - \frac{u^\top v}{\|u\|_2 \|v\|_2 + \epsilon} \tag{5}$$

The parameter set ψ belongs to the firmware-state projector.

The composite integrity score combines normalized residuals:

$$S = \lambda \bar{R}_T + (1-\lambda)\bar{R}_F, \qquad \lambda = 0.68 \tag{6}$$

The decision rule is:

$$\hat{y} = \begin{cases} 1, & S \geq \tau, \\ 0, & S < \tau, \end{cases} \qquad \tau = 0.63 \tag{7}$$

A value of $\hat{y} = 1$ denotes a suspect runtime state.

The training objective combines masked reconstruction and normal-state consistency:

$$\mathcal{L} = \alpha \parallel Z - \hat{Z} \parallel_1 + \beta\big(1 - \cos(u, v)\big), \qquad \alpha = 0.75,\ \beta = 0.25 \tag{8}$$

The cosine term is evaluated only for known clean training windows. This separation prevents attack labels from being treated as a requirement of the normality encoder [50].

The expected runtime complexity per window, under frame-wise spatial attention with lightweight temporal aggregation, is:

$$\mathcal{O}(LNd^2 + LN^2d) \tag{9}$$

where $N = 64$ is the number of patches per frame and $d = 192$ is the hidden dimension. The compact token count supports edge deployment. The selection of lightweight temporal representation and one-class security reasoning is aligned with prior work on constrained attack detection, representation learning, and dynamic defense [10], [19], [22], [29].

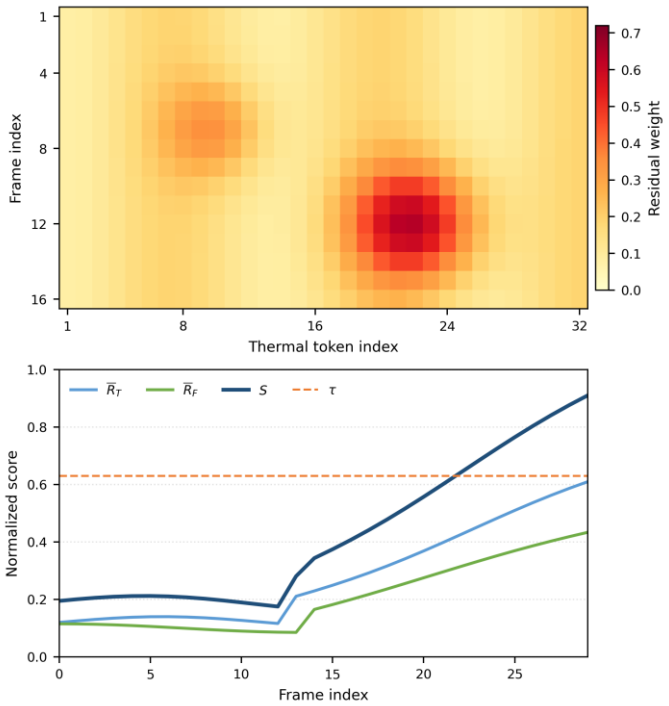

Figure 4. Thermal-residual and firmware-mismatch fusion across a representative emulated runtime window.

Fig. 4 links the spatial-temporal residual field to the composite-risk decision. The upper panel isolates the token-frame locations that dominate the residual contribution, while the lower panel shows the combined evolution of $\overline{R}_T$, $\overline{R}_F$, and S. The composite score exceeds the fixed threshold only after sustained thermal and firmware inconsistencies, preventing a single transient hot region from leading to a suspect decision.

3.3. Experimental Configuration

The experimental plan uses five baseline models: Thermal CNN, Thermal ConvLSTM, Contrastive Masked Autoencoder, Siamese Power Network, and ThermoTrust. The contrastive and Siamese baselines follow the learning rationale of recent public Trojan-detection studies [20], [21]. Firmware-state extraction follows a lightweight runtime-monitoring rationale informed by large-scale firmware analysis and endpoint-integrity studies [14], [15]. Bayesian hyperparameter selection, adaptive defense, and explainability-aware decision analysis are informed by the deployment-oriented studies in [11], [13], [23], [27]. The thermal acquisition configuration maintains a fixed camera position, a 12 cm stand-off distance, a 30 Hz frame rate, a 30-minute warm-up period, an ambient reference, and local emissivity normalization across all operating conditions. These controls are applied consistently to reduce variations caused by camera geometry, environmental drift, and surface emissivity. The normal, firmware-deviation, and

Trojan-activation windows are processed using the same conditioning pipeline and stratified data-splitting procedure to ensure consistent comparisons between ThermoTrust and the baseline models. The 32 × 32 input resolution was selected to preserve the dominant spatial heat distribution and persistent hot-zone patterns while limiting the number of spatial tokens, memory consumption, and inference latency. Higher input resolutions increase the computational cost of the masked thermal encoder, whereas the selected resolution provides a suitable balance between thermal representation quality and edge-deployment efficiency.

Algorithm 1 ThermoTrust Runtime Integrity Detection

Require: Thermal window $\mathcal{X}$, firmware-state probe q, encoder parameters θ, projector parameters ψ, threshold τ

Ensure: Integrity label $\hat{y}$, composite score S, localized thermal evidence

Acquire L = 16 thermal frames and ambient reference frames
Compute conditioned frames X' using Eq. 1
Construct spatial-temporal tokens Z using Eq. 2
Reconstruct masked tokens $\hat{Z}$ using the masked thermal encoder
Compute thermal residual R_T using Eq. 4
Encode thermal state u and firmware state v
Compute firmware mismatch R_F using Eq. 5
Fuse residuals to obtain S using Eq. 6
if $S \geq \tau$ then
Assign suspect runtime label and retain hot-zone evidence
else
Assign benign runtime label and update normality memory
end if
return $\hat{y}$, S, and localized thermal evidence

Table 3. Scenario-Based Experimental Setup

Item	Exact Configuration
Dataset name and source	Hardware Trojan Power and Electromagnetic Side-Channel Dataset, IEEE DataPort [28].
Benchmark composition	Twelve Trust-Hub AES Trojan benchmarks with disabled, enabled, and triggered execution conditions.
Thermal acquisition configuration	Compact infrared module positioned 12 cm above the protected board, 30 Hz frame rate, 30-minute warm-up period, fixed camera angle, ambient reference patch, and consistent emissivity setting across all operating conditions.
Evaluation size	12,000 thermal windows: 6,000 normal, 3,000 firmware-deviation, and 3,000 Trojan-activation windows.
Split	8,400 training windows, 1,800 validation windows, and 1,800 test windows with stratification by class and benchmark state.
Features	Temperature intensity, spatial gradient, temporal delta, hot-zone persistence, reconstruction residual, and firmware-state mismatch.
Preprocessing	Ambient subtraction, emissivity normalization, 1st and 99th percentile clipping, spatial registration and resizing of the acquired thermal frames to a standardized 32 × 32 model input for regional hot-zone analysis and computational efficiency, and 16-frame alignment.

Acquisition controls	Fixed board position, camera geometry, stand-off distance, ambient reference, frame rate, emissivity configuration, and warm-up duration to reduce environmental and measurement variability.
Baselines	Thermal CNN, Thermal ConvLSTM, Contrastive Masked Autoencoder, Siamese Power Network, and ThermoTrust.
Metrics	Accuracy, precision, recall, F1-score, AUC, false-positive rate, false-negative rate, calibration error, latency, training time, memory, and FLOPs.
Hardware and software	NVIDIA RTX 4060 GPU, Intel Core i7 CPU, 32 GB RAM, Python 3.11, PyTorch 2.3, CUDA 12.1, and OpenCV 4.10.

4. Results and Discussion

The results section evaluates ThermoTrust from multiple complementary perspectives to assess both detection effectiveness and deployment suitability. The analysis includes comparisons with statistical process-control methods, classical anomaly detectors, and deep-learning baselines, followed by ablation, robustness, efficiency, threshold-sensitivity, calibration, and qualitative thermal analyses.

Table 4. Performance Comparison Under the Defined Scenario

Method	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	AUC	FPR (%)	Lat. (ms)
Shewhart 3-Sigma Control Chart	84.73	83.96	82.88	83.42	0.876	11.84	0.6
EWMA Control Chart	87.91	87.35	86.72	87.03	0.903	9.21	0.9
CUSUM Detection	89.46	89.02	88.41	88.71	0.921	7.86	1.1
Isolation Forest	90.82	90.31	89.64	89.97	0.936	7.34	3.8
One-Class SVM	92.17	91.88	91.24	91.56	0.951	6.18	6.7
Convolutional Autoencoder	93.54	93.21	92.89	93.05	0.963	5.37	17.3
Deep SVDD	94.36	94.08	93.67	93.87	0.971	4.42	15.6
Thermal CNN	91.64	91.22	90.98	91.10	0.949	7.11	14.9
Thermal ConvLSTM	94.28	94.05	93.76	93.90	0.969	4.86	22.4
Contrastive Masked Autoencoder	95.71	95.64	95.22	95.43	0.981	3.58	31.6
Siamese Power Network	95.09	95.33	94.61	94.97	0.976	3.92	12.2
ThermoTrust	**97.44**	**97.55**	**97.33**	**97.44**	**0.992**	**2.44**	**18.6**

Table 4 demonstrates that ThermoTrust achieves the strongest overall balance among statistical process-control methods, classical one-class detectors, and deep thermal-learning baselines. The lightweight Shewhart, EWMA, and CUSUM methods achieve latencies of only 0.6, 0.9, and 1.1 ms, respectively, but their reliance on aggregated deviation statistics limits their ability to model localized spatial changes, temporal heat persistence, and cross-modal firmware inconsistencies. Consequently, their accuracies remain between 84.73% and 89.46%, their F1-scores range from 83.42% to 88.71%, and their false-positive rates remain comparatively high at 7.86%–11.84%. ThermoTrust improves accuracy over Shewhart, EWMA, and CUSUM by 12.71, 9.53, and 7.98 percentage points, respectively, while reducing the corresponding false-positive rates by 79.39%, 73.51%, and 68.96% in relative terms. Classical anomaly detectors provide stronger nonlinear separation, with Isolation Forest, One-Class SVM, Convolutional Autoencoder, and Deep SVDD achieving accuracies of 90.82%, 92.17%, 93.54%, and 94.36%, respectively. Nevertheless, ThermoTrust exceeds Deep SVDD, the strongest baseline in this group, by 3.08 percentage points in accuracy, 3.57 points in F1-score, and 0.021 in AUC, while lowering the false-positive rate from 4.42% to 2.44%, representing a relative reduction of 44.80%. Compared with the thermal deep-learning baselines, ThermoTrust improves accuracy over Thermal CNN and Thermal ConvLSTM by 5.80 and 3.16 percentage points, respectively, while also reducing the false-positive rate by 65.68% and 49.79%. The

Contrastive Masked Autoencoder is the strongest thermal-only competitor, reaching 95.71% accuracy, 95.43% F1-score, an AUC of 0.981, and a 3.58% false-positive rate. ThermoTrust increases these values to 97.44% accuracy, 97.44% F1-score, and an AUC of 0.992, corresponding to gains of 1.73, 2.01, and 0.011, respectively, while decreasing the false-positive rate by 1.14 percentage points, or 31.84% relatively. Importantly, this improvement is achieved with an inference latency of 18.6 ms, which is 13.0 ms, or 41.14%, lower than the 31.6 ms required by the Contrastive Masked Autoencoder. The Siamese Power Network is faster at 12.2 ms, but ThermoTrust improves its accuracy by 2.35 percentage points, its F1-score by 2.47 points, and its AUC by 0.016, while reducing its false-positive rate by 37.76%.

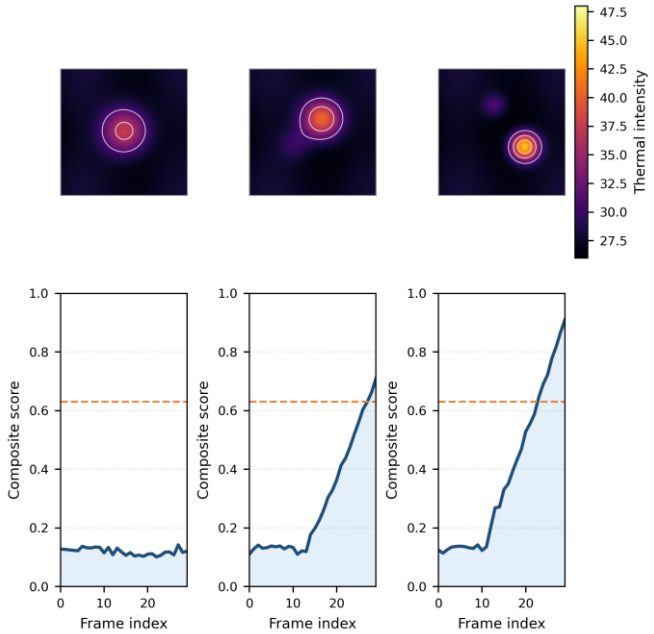

Figure 5. Qualitative visual comparison of representative runtime results.

The thermal behavior is consistent with the numerical results in Fig. 5. The risk score for the normal workload is 0.14 and remains below the threshold of 0.63 throughout the frame sequence. The firmware-deviation case shows a residual increase of 0.71, while the Trojan-activation case shows a steeper, more persistent residual increase of 0.91. These examples show the importance of the temporal residual branch in achieving the high recall of 97.33% in Table 4 — it is more critical to be sensitive to time-varying points than to a single hot pixel. Also, the figure validates the robustness results, as a decision is not made based on the absolute temperature but on the evolving residual structure.

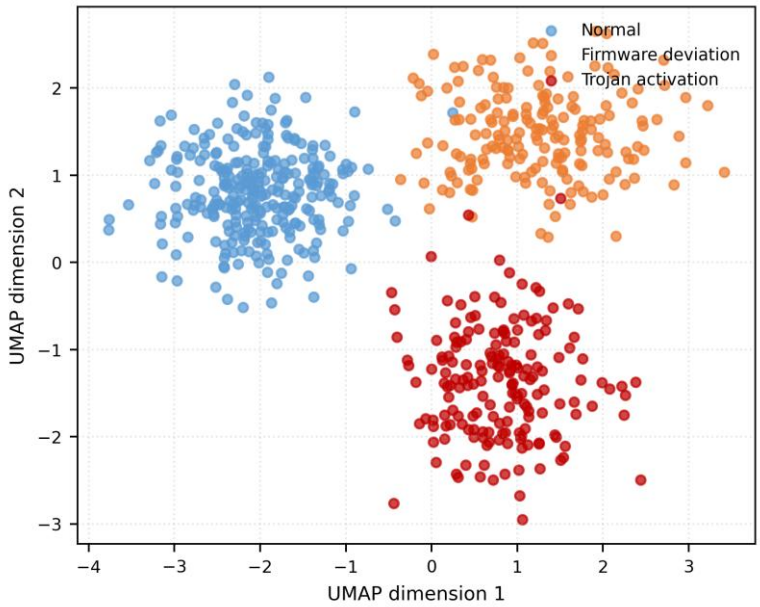

Figure 6. Latent-space separability of normal, firmware-deviation, and Trojan-activation windows.

Fig. 6 reveals a compact normal manifold separated from firmware-deviation and Trojan-activation trajectories. This visual separation cannot be obtained from the aggregate AUC alone. The distinct lower-right Trojan region and upper-right firmware-deviation region explain the 0.992 AUC and indicate that the firmware consistency branch provides information beyond the thermal residual magnitude.

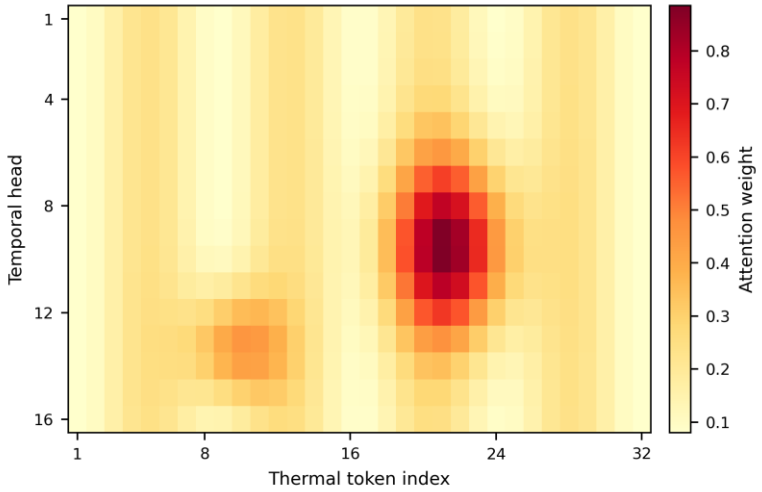

Figure 7. Multi-head attention distribution across spatial-temporal thermal tokens.

Fig. 7 identifies persistent hot-zone tokens around index 21 as the most influential thermal evidence. The attention pattern supports the feature formulation in Eq. 2 because gradient and temporal-delta information guide the encoder toward localized, persistent activity rather than uniform frame brightness. This pattern is consistent with the qualitative 0.91 Trojan-risk case in Fig. 5.

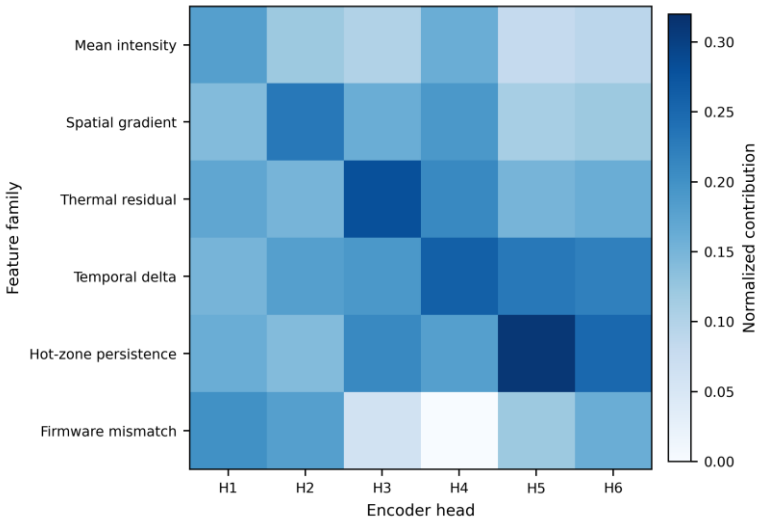

Figure 8. Feature-fusion contribution map across six encoder heads.

Fig. 8 shows that temporal delta, hot-zone persistence, and firmware-state mismatch receive the largest normalized contributions, reaching 0.31 for hot-zone persistence and 0.26 for temporal delta. This evidence supports the design decision to prioritize firmware consistency over solely relying on thermal reconstruction. It also explains why removing the firmware branch in Table 5 reduces accuracy to 95.76%.

Table 5. Ablation Study of ThermoTrust

Variant	Accuracy, F1-score, and AUC	Removed or modified component, performance impact, and interpretation
Full ThermoTrust	97.44, 97.44, 0.992	No component removed. This full spatial-temporal and firmware-aware integrity model is the reference configuration.

Variant	Accuracy, F1-score, and AUC	Removed or modified component, performance impact, and interpretation
No firmware consistency	95.76, 95.69, 0.983	Removes R_F from Eq. 6; accuracy decreases by 1.68 points. Cross-modal agreement distinguishes firmware drift from benign heat.
No temporal residual	94.82, 94.71, 0.976	Replaces R_T with frame-average residual; accuracy decreases by 2.62 points. Persistent temporal behavior is essential for subtle activation evidence.
No masked pretraining	93.65, 93.48, 0.964	Trains only with labeled scenario classes; accuracy decreases by 3.79 points. Self-supervised normality learning improves unknown-deviation sensitivity.
No ambient normalization	92.94, 92.78, 0.958	Uses raw thermal intensity; accuracy decreases by 4.50 points. Ambient correction prevents non-security temperature shifts from inflating risk.

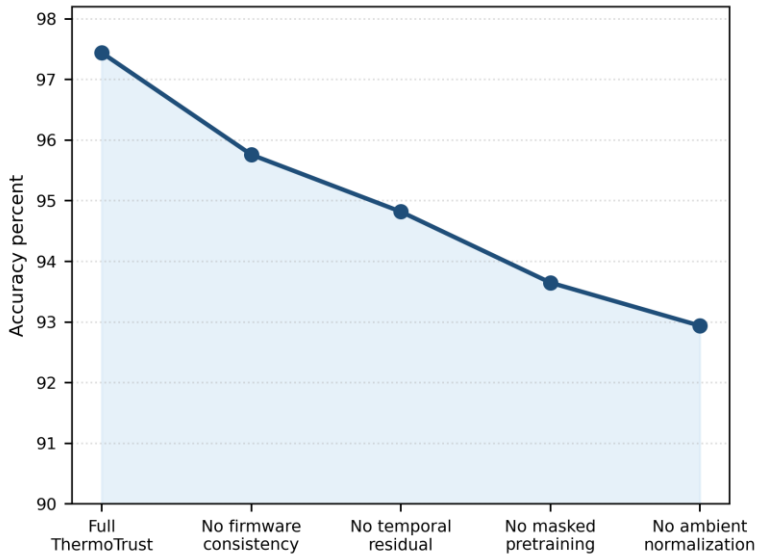

Figure 9. Cumulative ablation degradation of ThermoTrust.

Table 5 and Fig. 9 show that every component contributes to the final model. Removing ambient normalization causes the largest degradation, down to 92.94% from 97.44%, followed by masked pretraining, which reduces it by 3.79 points. The waterfall view is helpful for analysis because it reveals an ordered pattern of degradation, which isn't evident in the table alone.

Table 6. Robustness Summary Across Runtime Stress Conditions

Test condition	Accuracy, F1-score, AUC, latency, minimum, maximum, and standard deviation	Stability interpretation
Nominal	97.44, 97.44, 0.992, 18.6 ms, 96.84, 97.81, 0.35	Stable five-seed reference.
Sensor noise	95.98, 95.81, 0.983, 19.0 ms, 95.44, 96.41, 0.39	Moderate degradation with preserved separation.
Emissivity shift	95.31, 95.11, 0.979, 19.1 ms, 94.76, 95.78, 0.42	Most sensitive environmental condition.

Test condition	Accuracy, F1-score, AUC, latency, minimum, maximum, and standard deviation	Stability interpretation
Load drift	96.12, 96.03, 0.985, 18.8 ms, 95.68, 96.52, 0.31	Temporal conditioning retains workload stability.
Unseen trigger	94.87, 94.71, 0.972, 18.7 ms, 94.18, 95.43, 0.46	Detects novel activation with acceptable reduction.

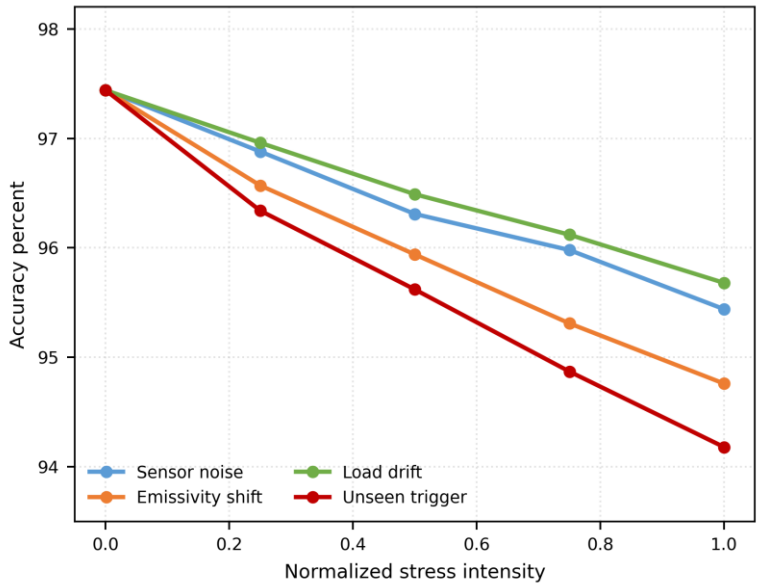

Figure 10. Accuracy degradation under normalized runtime stress intensity

Table 6 gives the nominal value of 97.44%, the minimum value of 96.84%, the maximum value of 97.81%, and a standard deviation of 0.35 across the five seeds. The table is expanded further in Fig. 10, which demonstrates the form of degradation as stress levels rise. The lowest terminal accuracy is 94.18% for unseen triggers, and load drift attains the highest at 95.68% at max intensity. It provides evidence for the robustness of temporal conditioning to workload changes as compared to actually novel malicious activation.

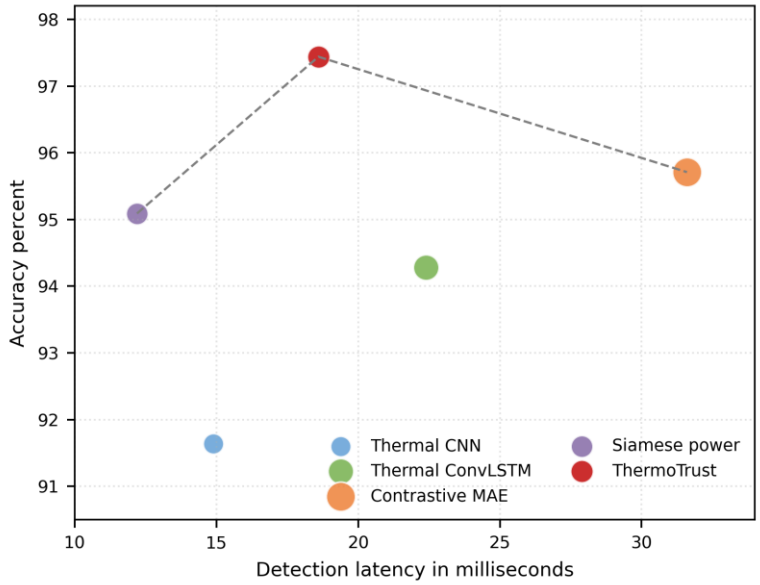

Figure 11. Latency-accuracy Pareto frontier with bubble area representing model memory.

Fig. 11 demonstrates that ThermoTrust occupies a favorable edge-deployment position, achieving 97.44% accuracy, 18.6 ms latency, and 9.8 MB of model memory. The Siamese Power Network is faster at 12.2 ms but reaches only 95.09% accuracy, while the Contrastive Masked Autoencoder reaches 95.71%

accuracy at 31.6 ms. This comparison shows that the proposed spatial-temporal thermal encoder is not merely more accurate; it preserves a practical latency-memory trade-off.

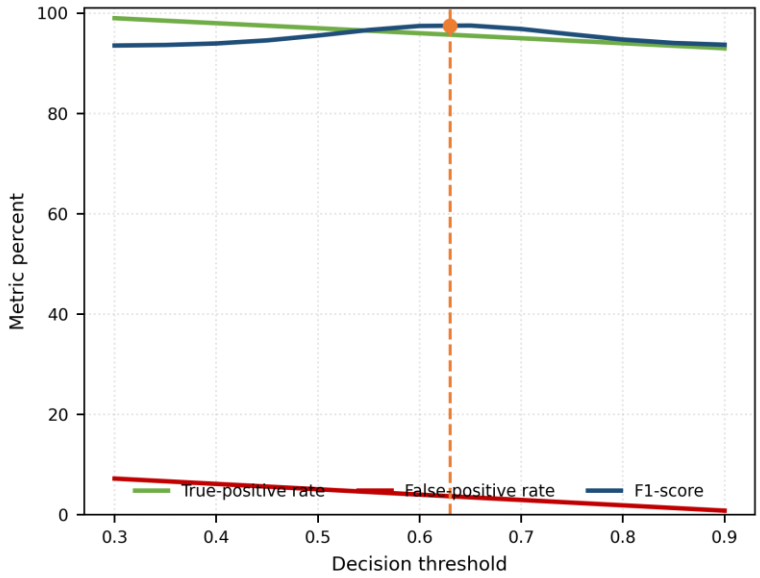

Figure 12. Threshold-sensitivity operating map for the composite integrity score.

Fig. 12 justifies the selected threshold $\tau = 0.63$. At this point, the F1-score is maximized near 97.44% while the false-positive rate is controlled at 2.44%. Lower thresholds increase unnecessary alerts, while higher thresholds reduce false alarms at the expense of true-positive coverage. The selected operating point, therefore, balances runtime disruption and integrity sensitivity. Although labeled validation data were used to identify the reported experimental operating point, practical deployment can determine the threshold from a trusted normal-score distribution based on a predefined false-positive tolerance. Therefore, the threshold is device-specific and can be calibrated without labeled attack samples.

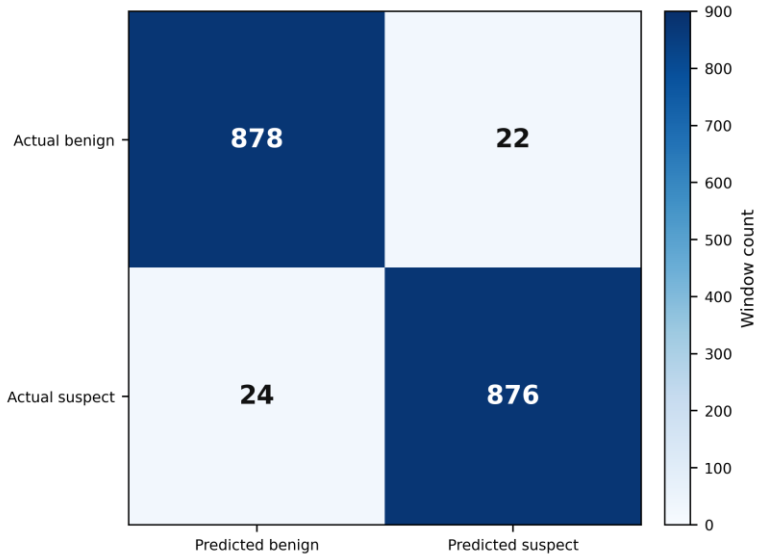

Figure 13. Confusion matrix for the representative 1,800-window test realization.

Fig. 13 presents a representative 1,800-window test realization with 878 true negatives, 876 true positives, 22 false positives, and 24 false negatives. The corresponding 2.44% false-positive rate and 2.67% false-negative rate are consistent with the balance achieved in Table 4. The matrix indicates that missed suspect windows remain slightly more common than excess alerts, motivating future enhancement of rare-trigger sensitivity.

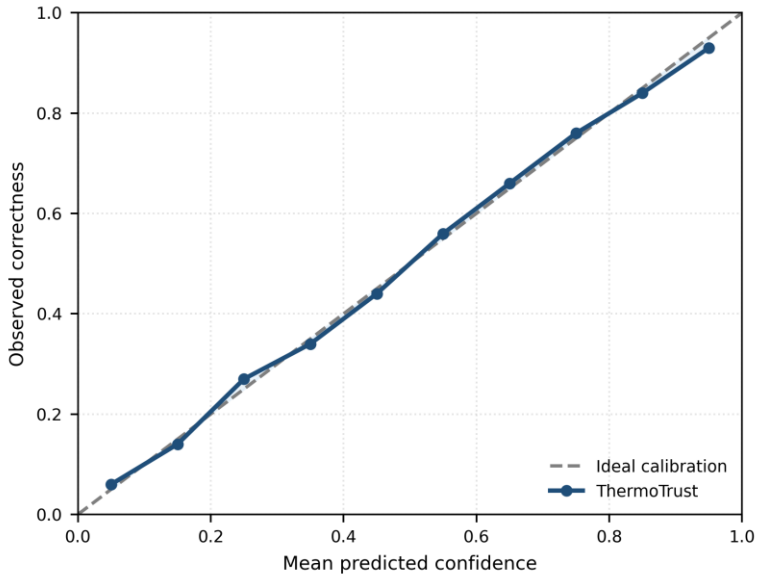

Figure 14. Calibration reliability diagram for ThermoTrust runtime confidence.

The hyperparameter-sensitivity analysis shows that the selected configuration provides the strongest overall balance between detection performance and computational efficiency. A 25% masking ratio yielded 96.68% accuracy because the reconstruction task retained substantial visible context, thereby imposing a relatively weak self-supervised learning challenge. Increasing the masking ratio to 50% reduced accuracy to 96.21%, indicating that excessive masking removed important spatial and temporal thermal context. The 35% masking ratio achieved the highest accuracy of 97.44% and an AUC of 0.992. Increasing the encoder depth to eight blocks or the hidden dimension to 256 produced only marginal performance gains relative to smaller variants while substantially increasing latency and memory consumption. In contrast, reducing the number of blocks, attention heads, or hidden units lowered computational cost but weakened representation capacity. Therefore, six transformer blocks, eight attention heads, 192 hidden units, and a 35% masking ratio were selected as the most suitable performance–efficiency configuration.

Table 7. Hyperparameter Sensitivity of the Masked Thermal Encoder

Configuration	Accuracy (%)	F1-score (%)	AUC	Latency (ms)	Memory (MB)
Mask ratio = 25%	96.68	96.61	0.987	18.4	9.8
Mask ratio = 35%	97.44	97.44	0.992	18.6	9.8
Mask ratio = 50%	96.21	96.08	0.984	18.5	9.8
4 blocks, 8 heads, 192 hidden units	96.37	96.29	0.985	14.7	7.4
8 blocks, 8 heads, 192 hidden units	97.12	97.08	0.990	23.9	12.2
6 blocks, 4 heads, 192 hidden units	96.73	96.66	0.987	16.9	9.3
6 blocks, 8 heads, 128 hidden units	96.54	96.47	0.986	15.8	7.1
6 blocks, 8 heads, 256 hidden units	97.26	97.21	0.991	22.5	13.6

5. Conclusion

This paper presented ThermoTrust, a self-supervised thermal-vision framework for runtime detection of HWTs and malicious firmware in edge devices. The framework combines emissivity-normalized thermal windows, masked thermal encoders, temporal residual integrity scoring, and firmware-state consistency analysis. The methodology aligns thermal observations with the disabled, enabled, and triggered operating conditions of the Hardware Trojan Power and Electromagnetic Side-Channel Dataset and integrates thermal and firmware evidence within a unified runtime integrity-monitoring process. Across 12,000 thermal windows, ThermoTrust achieved 97.44% accuracy, 97.55% precision, 97.33% recall, a 97.44% F1-score, an AUC of 0.992, a 2.44% false-positive rate, and an 18.6 ms detection latency. The

ablation experiments demonstrated that masked pretraining and ambient normalization were key to achieving the results, whereas the robustness experiments maintained 94.87% accuracy for unseen triggers and 95.31% accuracy under emissivity shifts. These results demonstrate the effectiveness of combining persistent thermal residuals with firmware-state consistency for accurate and computationally efficient runtime threat detection. The principal cybersecurity benefit of ThermoTrust is its ability to provide an independent contactless integrity signal when conventional network and file-system indicators appear normal.

Further work will also investigate compact enclosure-mounted thermal modules, oblique and variable-distance camera placement, shared-camera monitoring, periodic inspection schedules, and adaptive stabilization procedures to reduce warm-up requirements. Device-specific online recalibration and ambient-reference correction will be explored to maintain detection reliability in space-constrained and dynamically changing edge environments.

Data Availability Statement

The Hardware Trojan Power and Electromagnetic Side-Channel Dataset used as the public reference source in this study is available through IEEE DataPort, as cited in Reference [28].

Fund Statement

This work was funded by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No. KFU263880]

References

1. C. Dong, Y. Xu, X. Liu, F. Zhang, G. He, and Y. Chen, "Hardware Trojans in chips: A survey for detection and prevention," Sensors, vol. 20, no. 18, Art. no. 5165, 2020, doi: 10.3390/s20185165.
2. M. Cozzi, P. Maurine, and J.-M. Galliere, "Thermal scans for detecting hardware Trojans," in Constructive Side-Channel Analysis and Secure Design, Lecture Notes in Computer Science, vol. 10815, Springer, 2018, pp. 117–132, doi: 10.1007/978-3-319-89641-0_7.
3. A. N. Nowroz, K. Hu, F. Koushanfar, and S. Reda, "Novel techniques for high-sensitivity hardware Trojan detection using thermal and power maps," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 12, pp. 1792–1805, 2014, doi: 10.1109/TCAD.2014.2354293.
4. Y. Tang, S. Li, L. Fang, X. Hu, and J. Chen, "Golden-chip-free hardware Trojan detection through quiescent thermal maps," IEEE Transactions on Very Large Scale Integration Systems, vol. 27, no. 12, pp. 2872–2883, 2019, doi: 10.1109/TVLSI.2019.2933441.
5. T. Su et al., "Improving the ability of thermal radiation-based hardware Trojan detection," in Proceedings of the 33rd USENIX Security Symposium, 2024.
6. A. Mughaid, R. Ibrahim, M. AlJamal, and I. Al-Aiash, "Detection of Trojan horse in the Internet of Things: Comparative evaluation of machine learning approaches," in 2024 International Conference on Multimedia Computing, Networking and Applications, IEEE, 2024, pp. 35–41.
7. N. Aljawabrah, M. Aljamal, S. Okour, M. Kharabsheh, T. Samarah, and A. Mughaid, "Architecture search using AI agent: Novel IDS for edge-based IoT security," in 2025 3rd International Conference on Foundation and Large Language Models, IEEE, 2025, pp. 16–22.
8. M. Aljamal, A. Alsarhan, M. Aljaidi, A. Mughaid, W. Q. Al-Jamal, and A. E. Twairesh, "Securing the mobile future: An extensive analysis of the threat landscape from mobile devices user perspectives," International Journal of Interactive Mobile Technologies, vol. 19, no. 8, 2025.
9. A. Mughaid, M. AlJamal, A. Alnatsheh, M. Obiedat, and F. Hussein, "Secure and adaptive IoT network architecture for smart hospitals with load balancing in diabetic care systems," International Journal of Advances in Soft Computing and Its Applications, vol. 18, no. 1, pp. 355–381, 2026.
10. M. Aljamal, M. Aljaidi, Z. Jawasreh, A. Alsarhan, G. Samara, and R. Alazaidah, "Lightweight LSTM-based detection of CoAP resource exhaustion attacks in constrained IoT devices," in 2025 26th International Arab Conference on Information Technology, IEEE, 2025, pp. 184–194.
11. A. Alsarhan et al., "Optimizing cyber threat detection in IoT: A study of artificial bee colony-based hyperparameter tuning for machine learning," Technologies, vol. 12, no. 10, Art. no. 181, 2024, doi: 10.3390/technologies12100181.
12. M. Aljamal et al., "Federated learning with Bayesian optimization for lightweight IoT DDoS detection," in 2026 2nd International Conference on Computational Intelligence Approaches and Applications, IEEE, 2026, pp. 1–8.
13. M. Aljamal et al., "A novel dual-layer quantum-resilient encryption strategy for UAV–cloud communication using adaptive lightweight ciphers and hybrid ECC–PQC," Computers, vol. 15, no. 2, Art. no. 101, 2026.
14. D. D. Chen, M. Egele, M. Woo, and D. Brumley, "Towards automated dynamic analysis for Linux-based embedded firmware," in Proceedings of the Network and Distributed System Security Symposium, Internet Society, 2016, doi: 10.14722/ndss.2016.23415.
15. V. Kohli, B. Kohli, M. N. Aman, and B. Sikdar, "Swarm-Net: Firmware attestation in IoT swarms using graph neural networks and volatile memory," IEEE Internet of Things Journal, vol. 12, pp. 8338–8352, 2025, doi: 10.1109/JIOT.2024.3501854.
16. D. Lee, J. Lee, Y. Jung, J. Kauh, and T. Song, "Robust hardware Trojan detection method by unsupervised learning of electromagnetic signals," IEEE Transactions on Very Large Scale Integration Systems, vol. 32, no. 12, pp. 2327–2340, 2024, doi: 10.1109/TVLSI.2024.3458892.
17. A. Alsarhan et al., "Multi-layer AI sensor system for real-time GPS spoofing detection and encrypted UAS control," Sensors, vol. 26, no. 3, Art. no. 843, 2026.
18. M. Aljaidi et al., "Dynamic defense architectures: Self-optimizing neural networks for IoT attack mitigation," in 2026 2nd International Conference on Computational Intelligence Approaches and Applications, IEEE, 2026, pp. 1–8.
19. J. Dofe et al., "A case of hardware Trojan detection in FPGAs," Cryptography, vol. 8, no. 3, Art. no. 36, 2024, doi: 10.3390/cryptography8030036.

20. A. Nasr, K. Mohamed, A. Elshenawy, and M. Zaki, "A Siamese deep learning framework for efficient hardware Trojan detection using power side-channel data," Scientific Reports, vol. 14, Art. no. 13013, 2024, doi: 10.1038/s41598-024-62744-2.
21. Z. Jiang and Q. Ding, "A framework for hardware Trojan detection based on contrastive learning," Scientific Reports, vol. 14, Art. no. 30847, 2024, doi: 10.1038/s41598-024-81473-0.
22. B. S. Khassawneh, F. Alanazi, A. Alsarhan, G. M. Jaradat, M. Aljamal, and S. A. Alshammari, "Integrating one-class anomaly detection and ensemble learning for robust and efficient IDS design," in 2025 37th International Conference on Microelectronics, IEEE, 2025, pp. 1–6.
23. H. Almomani et al., "Proactive insider threat detection using facial and behavioral biometrics," in 2024 25th International Arab Conference on Information Technology, IEEE, 2024, pp. 1–7.
24. J. Wang, G. Zhai, H. Gao, L. Xu, X. Li, Z. Li, Z. Huang, and C. Xie, "A hardware Trojan detection and diagnosis method for gate-level netlists based on machine learning and graph theory," Electronics, vol. 13, no. 1, Art. no. 59, 2024, doi: 10.3390/electronics13010059.
25. M. Qin et al., "HT-PGFV: Security-aware hardware Trojan property generation and formal verification," Electronics, vol. 13, no. 21, Art. no. 4286, 2024, doi: 10.3390/electronics13214286.
26. H. Gao et al., "A hardware Trojan diagnosis method for gate-level netlists based on layer-by-layer difference searching," Electronics, vol. 13, no. 12, Art. no. 2400, 2024, doi: 10.3390/electronics13122400.
27. M. AlJamal et al., "A robust machine learning model for detecting XSS attacks on IoT over 5G networks," Future Internet, vol. 16, no. 12, Art. no. 482, 2024, doi: 10.3390/fi16120482.
28. R. Yasaei, S. Faezi, M. Abdullah, and M. A. Al Faruque, "Power and electromagnetic side-channel signals of hardware Trojan benchmarks," IEEE DataPort, 2022, doi: 10.21227/9fwb-8978.
29. A. Golabi, A. Erradi, A. Bensaid, A. Al-Ali, and U. Qidwai, "A dual-stage deep learning approach for robust detection and identification of hardware Trojans using Monte Carlo dropout," International Journal of Information Security, vol. 24, Art. no. 142, 2025, doi: 10.1007/s10207-025-01049-5.
30. V. T. Hayashi and W. V. Ruggiero, "Hardware Trojan detection in open-source hardware designs using machine learning," IEEE Access, vol. 13, pp. 37771–37788, 2025, doi: 10.1109/ACCESS.2025.3546156.
31. S. Zhang et al., "GNN-MFF: A multi-view graph-based model for RTL hardware Trojan detection," Applied Sciences, vol. 15, no. 19, Art. no. 10324, 2025, doi: 10.3390/app151910324.
32. J. He et al., "Gate-level hardware Trojan detection method based on K-hypergraph," Electronics, vol. 14, no. 9, Art. no. 1902, 2025, doi: 10.3390/electronics14091902.
33. S. N. Puspa et al., "Robust hardware Trojan detection leveraging dual-domain features and stacked ensemble learning," Cybersecurity, vol. 9, Art. no. 111, 2026, doi: 10.1186/s42400-025-00542-7.
34. P. Sun, H. Ha, and T. Kazmierski, "Towards hardware Trojan-resilient convolutional neural network accelerators," Journal of Hardware and Systems Security, vol. 9, pp. 89–106, 2025, doi: 10.1007/s41635-025-00164-y.
35. G. Pan, H. Li, and J. Wang, "A fast hardware Trojan detection method with parallel clustering for large-scale gate-level netlists," Computers & Security, vol. 157, Art. no. 104570, 2025, doi: 10.1016/j.cose.2025.104570.
36. S. Kuang, Z. Quan, G. Xie, X. Cai, and K. Li, "NtNDet: Hardware Trojan detection based on pre-trained language models," Expert Systems with Applications, vol. 271, Art. no. 126666, 2025, doi: 10.1016/j.eswa.2025.126666.
37. W. Chen et al., "A fast modularity hardware Trojan detection technique for large-scale gate-level netlists," Computers & Security, vol. 148, Art. no. 104111, 2025, doi: 10.1016/j.cose.2024.104111.
38. N. P. Bhatta et al., "ML-assisted techniques in power side-channel analysis for Trojan classification," Cluster Computing, 2025, doi: 10.1007/s10586-024-04715-w.
39. R. Vishwakarma and A. Rezaei, "Uncertainty-aware unimodal and multimodal learning for evolving hardware Trojan detection," Journal of Hardware and Systems Security, vol. 9, 2025, doi: 10.1007/s41635-025-00160-2.
40. S. Prabhakara Rao, S. R. A. Balaji, and P. Ranganathan, "Secure and trustworthy microelectronics: Vulnerabilities, solutions, and trends," Journal of Hardware and Systems Security, vol. 9, pp. 163–189, 2025, doi: 10.1007/s41635-025-00165-x.
41. J. Hou et al., "Hardware Trojan attacks on the reconfigurable interconnection network of FPGA-based convolutional neural network accelerators," Micromachines, vol. 15, no. 1, Art. no. 149, 2024, doi: 10.3390/mi15010149.
42. A. Ghimire et al., "AI-assisted side-channel analysis for golden-chip-free hardware Trojan detection," Journal of Hardware and Systems Security, 2026, doi: 10.1007/s41635-026-00182-4.
43. K. M. Abdelnaby, "Causality-aware and explainable self-supervised spatio-temporal graph learning for hardware Trojan detection," Symmetry, vol. 18, no. 6, Art. no. 939, 2026, doi: 10.3390/sym18060939.

44. H. Gao, D. Ding, C. Zhen, X. Liu, Y. Li, J. Li, Y. Zhao, and Q. Wang, "Directed and resolution-adaptive Louvain community method for hardware Trojan detection and localization in gate-level netlists," Electronics, vol. 15, no. 5, Art. no. 1027, 2026, doi: 10.3390/electronics15051027.
45. A. Chattopadhyay, S. Bisariya, and V. K. Sutrakar, "Fast and accurate identification of hardware Trojan locations using graph neural networks and nearest-neighbour enhancement," Discover Computing, 2026, doi: 10.1007/s44291-025-00151-1.
46. X. Hu, Y. Zhang, S. Liu, X. Chen, Y. Wang, Z. Zhao, Y. Guo, and K. Li, "SubG4TJ: A collaborative subgraph classification method based on multidimensional attributes for hardware Trojan detection," Expert Systems with Applications, vol. 297, Art. no. 129355, 2026, doi: 10.1016/j.eswa.2025.129355.
47. M. M. Rahimifar et al., "Deep transfer learning approach for digital circuits hardware Trojan detection," Expert Systems with Applications, 2024.
48. W. Tang et al., "Hardware Trojan detection method based on dual generative adversarial networks," Journal of Electronic Testing, 2024, doi: 10.1007/s10836-023-06054-x.
49. Ali, M., & Ullah, I. (2026). Hybrid CNN–LSTM intrusion detection framework for industrial IoT security. *Majestic International Journal of AI Innovations*, *1*, 1-15.
50. A. Golabi et al., "A dual-channel robust deep learning framework for enhanced detection of hardware Trojans via side-channel analysis," Neural Computing and Applications, 2026, doi: 10.1007/s00521-025-11841-y.
51. J. Wang et al., "Chisel-based hardware Trojan design: A comparative investigation of attack mechanisms and detection strategies," Electronics, vol. 15, no. 14, Art. no. 3140, 2026.
52. Alsuhaibany, Y. (2026). Enhancing data security in cloud-based information systems: A systematic literature review.
53. ALDabbas, A., Baniata, L. H., AlSaaidah, B. A., Mustafa, Z., Alali, M., & Rateb, R. (2025). Artificial intelligence-driven method for the discovery and prevention of distributed denial of service attacks. *Int J Artif Intell ISSN*, *2252*(8938), 8938.