IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 3, June 2026, pp. 2750∼2760
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i3.pp2750-2760

A transfer hybrid deep learning approach for advanced intrusion detection in IoT-based smart home security

Mouad Choukhairi, Ouail Choukhairi, Youssef Fakhri, Ali Choukri
LARI Laboratory, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

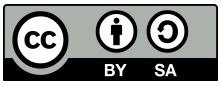
Received May 8, 2025
Revised Apr 1, 2026
Accepted Apr 22, 2026

Keywords:

Cybersecurity
Hybrid CNN-LSTM
Intrusion detection system
IoT smart home security
IoTID20 dataset
Particle swarm optimization
Transfer learning

ABSTRACT

As smart home environments increasingly rely on interconnected internet of things (IoT) devices, they face growing cyber threats originating both externally from malicious actors and internally from compromised or malfunctioning IoT devices. These threats, including unauthorized access, distributed denial of service (DDoS) attacks, and data exfiltration, pose significant risks to the security and privacy of smart home inhabitants. This paper introduces an advanced intrusion detection system (IDS) specifically designed to enhance the security of IoT-based smart home networks. Leveraging a hybrid deep learning approach combining convolutional neural networks (CNN) and long short-term memory (LSTM) models, complemented by transfer learning (TL) and hyper-parameter optimization techniques, our proposed IDS efficiently identifies both external and intra-network threats. Using the IoTID20 dataset, which simulates realistic attack scenarios, the IDS was trained and evaluated to detect abnormal behavior effectively within smart home networks. CNN layers extract spatial features from network traffic, while LSTM layers capture temporal dependencies, enabling robust detection against a range of cyber-threats. Evaluation results demonstrate the IDS's high detection accuracy and exceptional F1-scores, validating its effectiveness in safeguarding IoT-based smart homes from evolving threats.



Corresponding Author:

Mouad Choukhairi
LARI Laboratory, Department of Computer Science, Faculty of Sciences, Ibn Tofail University
B.P 133, University Campus, Kenitra, Morocco
Email: mouad.choukhairi@uit.ac.ma

1. INTRODUCTION

The pervasive integration of internet of things (IoT) devices into smart home environments has ushered in an era of unprecedented convenience and automation, introducing a complex web of interconnected devices that demand robust security measures [1]. These devices, ranging from smart thermostats and lighting systems to security cameras and personal assistants, have become integral components of modern living, enhancing convenience and efficiency while simultaneously exacerbating the cybersecurity landscape and introducing a plethora of novel threats, ranging from external intrusions orchestrated by malicious actors to internal compromises arising from malfunctioning IoT devices, pose significant risks to the confidentiality, integrity, and availability of smart home networks, as well as to the privacy of their inhabitants [2]. Moreover, attackers may target these vulnerabilities to gain unauthorized access to smart home networks, potentially compromising sensitive data, disrupting services, or even gaining control of physical devices. Common threats such as node capture, sniffing, malware, and boot-phase attacks compromise residents' privacy and interfere with normal

smart-home operations [3]. Furthermore, the constrained resources of IoT devices impede the implementation of robust security mechanisms, while their heterogeneity complicates the deployment of unified security solutions [4]. Traditional security mechanisms often prove inadequate in the face of these evolving threats, necessitating the development of an advanced intrusion detection system (IDS) capable of discerning subtle anomalies and sophisticated attack patterns [5]. In addition, they struggle to cope with the dynamic and heterogeneous nature of IoT-based smart home networks, often generating a high number of false positives and failing to detect sophisticated attacks [6]. The increasing volume of web traffic due to the proliferation of IoT devices has rendered traditional attack detection methods and outdated data processing techniques obsolete, highlighting the need for more advanced analytical approaches [7]. The limitations of conventional security measures in addressing the unique challenges posed by IoT environments underscore the urgent need for more sophisticated and adaptive IDS.

Existing research on intrusion detection for IoT-based smart homes has steadily progressed, yet important constraints remain. A recent survey by Liao *et al.* [8] synthesizes deep learning (DL) IDS efforts and highlights trade-offs between accuracy and memory footprint in edge deployments. In practice, Afroz *et al.* [9] demonstrate a machine-learning IDS called blockchain-assisted secure smart home network – gradient based optimizer with hybrid deep learning (BSSHN-GBOHDL) that attains 98.3% accuracy for smart-home traffic. Still, the authors note latency issues when the model is pushed to resource-constrained gateways. Karmous *et al.* [10] build an software-defined networking (SDN)-assisted framework that reaches 99.96% accuracy against man-in-the-middle (MITM) attacks; nonetheless, their study shows that additional controller overhead can impede real-time response as device counts grow. Together, these works confirm the need for IDS techniques that balance high detection rates with minimal computational cost. DL architectures have delivered notable gains by fusing spatial and temporal analysis of network traffic. Sinha *et al.* [11] propose a hybrid long short-term memory (LSTM) based on convolutional neural networks (CNN) model that captures both flow dynamics and packet-level features, achieving 99.87% accuracy on botnet of things (BoT)-IoT while maintaining a 0.13% false-positive rate. Complementing this, Nazir *et al.* [12] evaluate a CNN–LSTM pipeline across three IoT benchmarks IoT-23, N-BaIoT, CICIDS2017, and report near-perfect accuracy after applying principal component analysis (PCA) and model quantization to suit embedded devices. For purely convolutional designs, Deshmukh and Ravulakollu [13] introduce IIDNet, which incorporates feature selection and fine-tuned hyper-parameters, to reach 95.5% accuracy on UNSW-NB15, illustrating that streamlined CNNs can still perform competitively in smart-home contexts. Recent studies further show that transfer learning (TL) and automated hyper-parameter search significantly enhance IDS effectiveness. Abdelhamid *et al.* [14] integrate convolutional block attention module (CBAM) attention into pre-trained CNNs and fine-tune them on BoT-IoT data, improving detection while lowering training demands compared with training from scratch. In parallel, Qaddos *et al.* [15] employ particle swarm optimization (PSO) to tune a CNN - gated recurrent unit (GRU) hybrid and apply feature weighted - synthetic minority over-sampling technique (FW-SMOTE) balancing, achieving 99.6% accuracy on IoTID20 and 99.16% on UNSW-NB15, evidence that systematic hyper-parameter tuning and TL together can yield high accuracy with moderate resource use. Collectively, these studies underscore the importance of hybrid DL architectures, careful parameter optimization, and feature reuse for deploying high-fidelity IDS solutions in resource-limited smart-home environments.

To address these critical challenges and bridge the existing gaps in smart home security, our research introduces a novel IDS designed to enhance the security of IoT-based smart home networks. The proposed IDS leverages three hybrid DL variants, Inception-LSTM, Xception-LSTM, and InceptionResNet-LSTM, serving as the core learners, each trained and tested on smart-home traffic drawn from the IoTID20 benchmark dataset, which mirrors real attacks such as Mirai, scans, denial-of-service (DoS) floods, and MITM address resolution protocol (ARP) spoofing. Experiments show that the system reliably detects both external and internal threats, achieving high accuracy and F1-scores. The paper's main contributions are threefold: i) it introduces a unified end-to-end CNN-LSTM intrusion-detection framework capable of accurately identifying both external and intra-network attacks in IoT-based smart-home traffic; ii) it proposes a TL pipeline in which raw packet flows are transformed into image representations and processed by CNN-LSTM backbones pre-trained on large-scale vision datasets, with PSO employed to fine-tune critical hyperparameters and significantly enhance accuracy and generalization against evolving threats; and iii) it provides comprehensive validation on the IoTID20 benchmark, where extensive experiments under realistic smart-home scenarios demonstrate consistently high detection performance, confirming the practical effectiveness of the proposed IDS.

The rest of the paper is organized as follows: section 2 presents the architecture and key components of the proposed CNN-LSTM, data transformation, and TL-based IDS with PSO. Section 3 describes the experimental setup, including the experimental setup and the evaluation metrics used, and discusses the evaluation results and the implications of the findings. Finally, section 4 concludes the paper and outlines potential future research directions.

2. METHOD

This section details the end-to-end pipeline of the proposed IDS for IoT-based smart home networks. First, outline a multilayer architecture that positions lightweight detection logic on each device while maintaining a global view at the network gateway. Next, describe how raw packet captures are transformed into image-like tensors that preserve both spatial correlations among features and temporal ordering of flows. Finally, present the hybrid CNN–LSTM core, explain how TL and PSO jointly boost performance, and summarize the real-time detection workflow.

2.1. System architecture overview

The goal of this research is to design an IDS that can recognize a broad spectrum of cyber-attacks originating both inside an IoT-based smart-home network and from the external Internet gateway. Figure 1 illustrates a representative threat scenario together with the high-level architecture of an IDS-protected smart home. An adversary may i) compromise one or more connected domestic devices and pivot laterally across the home network, or ii) probe and flood the home gateway directly from outside the premises. To defend against these dual threat surfaces, the proposed IDS solution adopts a two-tier architecture:

– Device-level IDS module: a lightweight agent embedded in each IoT node (e.g. camera and thermostat) passively monitors outbound and inbound traffic, extracts a minimal set of statistical features, and forwards only feature vectors, rather than raw packets, to the gateway. This design respects memory/CPU limits while reducing privacy leakage [16].
– Gateway-level IDS module: a more resource-rich engine deployed on the residential router or edge server, this module ingests flow records from every device, correlates them with north-south traffic crossing the WAN interface, and executes global anomaly analysis [17].

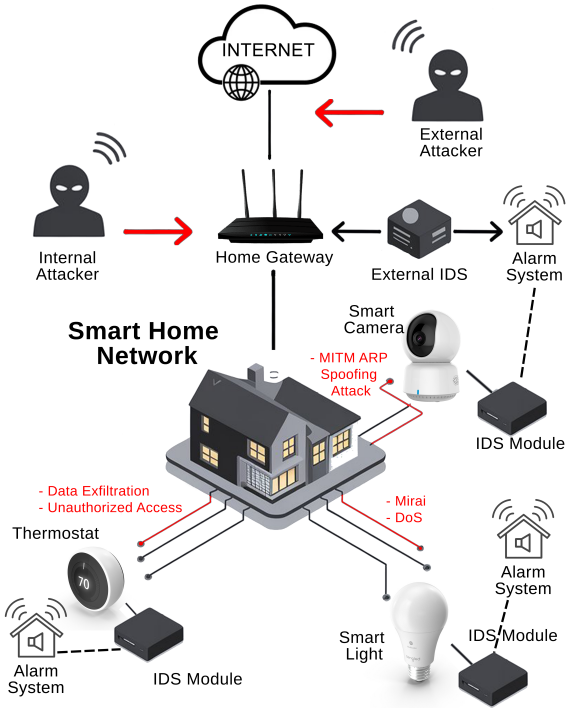


Figure 1. Illustrative cyber-attack pathway and its mitigation within an IDS-secured smart home

To protect the smart home network, the IDS follows four clear steps, as shown in Figure 2. First, it segments bidirectional traffic from each IoT device and the gateway into fixed, time-based windows. Second, every window is then normalized via a quantile transformation and reshaped into RGB-style tensors, ensuring that both spatial feature correlations and temporal flow dynamics are retained for subsequent DL analysis. Three, the resulting tensors form the training corpus for three transfer-learning backbones, Inception-LSTM, Xception-LSTM, and InceptionResNet-LSTM, each fine-tuned on the IoTID20 dataset. A PSO routine calibrates key hyper-parameters (i.e., learning rate, dropout, LSTM width, and epoch count) to maximize validation F1-score while respecting the computational constraints typical of embedded IoT platforms. Fourth, the IDS applies a decision rule to detect whether the traffic is normal or an attack, providing a low-latency, network-wide decision that effectively mitigates both internal and external threats.

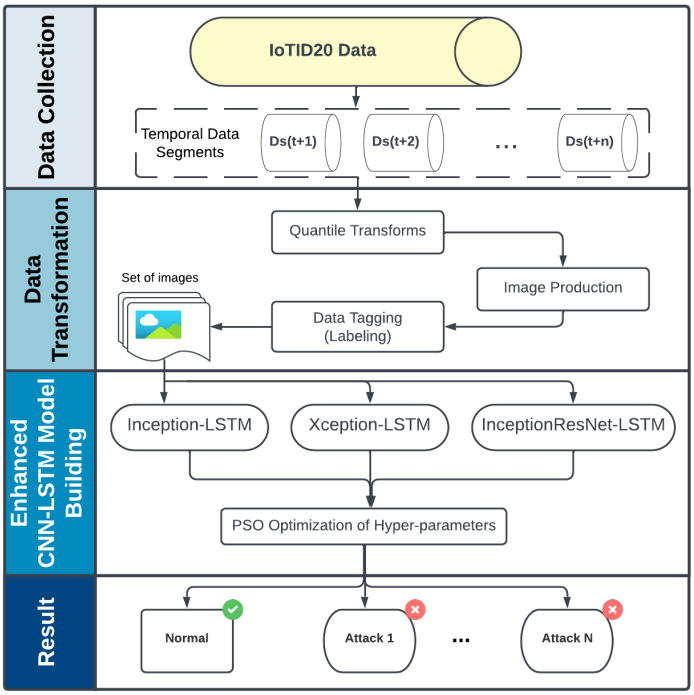


Figure 2. Adaptive IDS framework based on an optimized CNN–LSTM models

2.2. Data acquisition and representation

To rigorously evaluate the proposed IDS in both device-level and gateway-level scenarios, the IoTID20 is adopted, an open benchmark dataset, as a comprehensive testbed, because it captures both external and intra-network attacks typical of smart homes. It contains benign traffic as well as four representative attack families, such as Mirai, port scans, DoS floods, and MITM ARP spoofing. By reproducing device-to-device and internet-to-gateway flows, the dataset mirrors the operational realistic scenarios that a smart-home IDS must confront. As the raw capture is tabular (i.e., 76 flow-level features per packet), we transform it into an image-type format that exploits the spatial–temporal learning capacity of CNN–LSTM models. To prepare the data for hybrid DL modeling, the following representation pipeline is applied:

– Time-window chunking: for every device and gateway interface, consecutive packets are grouped into windows of 228×76 (i.e., rows$\times$features), yielding 17,328 values per window.
– Quantile normalization: each feature column is mapped to the interval $[0, 255]$ via a quantile transformation, which is more resilient to outliers than standard min–max scaling, while retaining relative rank information and producing a quasi-Gaussian distribution suitable for pixel-style inputs [18].
– Image reshaping: each window is folded row-wise into a square tensor of size $76 \times 76 \times 3$, producing an RGB image that encodes both feature correlations (i.e., spatial dimension) and packet sequencing (i.e., implicit temporal dimension) in a format optimized for CNN-LSTM processing.

Every resulting image is labeled by majority rule: normal if all 228 rows are benign, otherwise by the dominant attack class (e.g., Mirai or DoS). Figure 3 depicts a typical class representation. Normal traffic exhibits a uniform texture, Mirai flows display repetitive horizontal bands, scan attacks appear fragmented, DoS samples are dense and saturated, and MITM ARP spoofing reveals oscillatory streaks characteristic of deceptive address resolution. These visually distinct patterns confirm that the transformation preserves discriminative cues, enabling the CNN front-end to learn robust spatial features while the LSTM back-end layers capture temporal evolution (i.e., chronological order). The final balanced set of labeled images thus furnishes an appropriate domain for training, validating, and stress-testing the proposed hybrid IDS under realistic smart-home attack conditions.

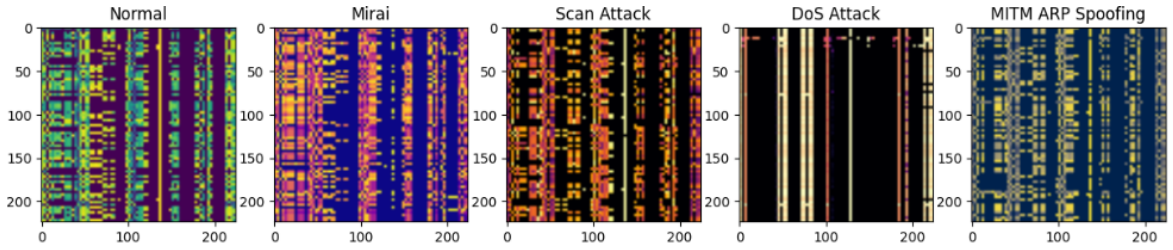


Figure 3. Class-wise image samples illustrating IoTID20's visual characteristics

2.3. CNN–LSTM based IDS core

Deep neural networks (DNN) have demonstrated superior performance over traditional machine learning techniques for network-security tasks, primarily because they can extract hierarchical features directly from raw data. In the present work, a hybrid DL architecture was employed, where convolutional layers learn spatial correlations in the image-like traffic tensors, whereas recurrent layers learn temporal dependencies within each flow window, yielding a detector that is sensitive to both instantaneous and evolving attack patterns [19], [20]. Algorithm 1 summarizes the complete workflow of the proposed IDS. It first outlines the off-line optimization of hyper-parameters via PSO, followed by the on-line inference workflow executed locally on each IoT node and collaboratively refined at the gateway.

Algorithm 1 Training and deployment of CNN-LSTM IDS

Require: IoTID20 traffic records $\mathcal{D}$, PSO hyper-parameter grid $\mathcal{H}$
Ensure: Optimized CNN–LSTM weights $\Theta^{\star}$
Off-line training phase
Segment $\mathcal{D}$ into windows of 228 packets
Apply quantile normalization and reshape each window to a tensor $X \in [0, 255]^{76\times76\times3}$
for all candidate hyper-parameter sets $h \in \mathcal{H}$ **via** PSO **do**
    Initialize CNN backbone with ImageNet weights
    Fine-tune convolutional layers specified by h
    Train the LSTM stack and dense classifier on $\mathcal{D}$ using h
    Evaluate validation $\mathrm{F}_1(h)$
end for
Select $h^{\star} = \arg\max_h \mathrm{F}_1(h)$ and save $\Theta^{\star}$
On-line deployment phase
for all incoming windows $\mathbf{X}_t$ on an IoT device **do**
    Compute spatial features with CNN$(\mathbf{X}_t; \Theta^{\star})$
    Update hidden state with LSTM
    Predict class $\hat{y}_t$ via soft-max
    if $\hat{y}_t \neq$ Normal **then**
        Forward alert score to gateway
    end if
end for

2.3.1. Convolutional front-end and temporal modeling

A pre-trained backbone (i.e., Inception, Xception, or InceptionResNet) receives a $76 \times 76 \times 3$ tensor and produces a high-level feature map $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$. By reusing ImageNet weights and fine-tuning only the upper layers, the network retains generic edge and texture detectors while adapting higher filters to traffic-specific motifs, thus accelerating convergence and reducing the risk of over-fitting in the limited IoTID20 domain. The feature map $\mathbf{F}$ is flattened into a sequence $\{\mathbf{f}_t\}_{t=1}^{T}$ and passed to a two-layer LSTM stack that captures long-range dependencies among packets within the time window. Let $(\mathbf{h}_t, \mathbf{c}_t)$ denote the hidden and cell states at step t; The LSTM update as in (1).

$$(\mathbf{h}_t, \mathbf{c}_t) = \mathrm{LSTM}(\mathbf{f}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\,;\, \Theta_{\mathrm{LSTM}}) \tag{1}$$

Where Θ_{LSTM} are trainable weights shared across time.

2.3.2. Decision layer and training

The final hidden state $\mathbf{h}_T$ is fed to a dense soft-max classifier that outputs posterior probabilities over the five classes, including Normal, Mirai, Scan, DoS, and MITM ARP. Model hyper-parameters, such as learning rate, drop-out, LSTM width and depth, and number of frozen CNN layers, are tuned via PSO to maximize validation F1-score while respecting embedded-device memory limits. During deployment, as seen in Figure 1, each IoT node performs local inference and forwards soft alerts to the gateway, where an ensemble decision combines device-level and network-wide evidence for a final decision. Therefore, this design unifies spatial feature extraction, temporal sequence modeling, and optimized hyper-parameter selection, providing a robust foundation for high-fidelity intrusion detection in resource-constrained smart-home environments.

2.4. Transfer learning and hyper-parameter optimization

The proposed framework incorporates TL and hyper-parameter optimization techniques to enhance IDS's performance and generalization capabilities. TL in DL models is a powerful technique involving the transfer of knowledge acquired by a DNN from one dataset (i.e., the source domain) to another dataset (i.e., the target domain), which is particularly useful when the target domain has little labeled data. Deep convolutional backbones pre-trained on large-scale natural-image corpora (e.g., ImageNet) have been shown to capture generic low-level filters (i.e., edges and textures) that remain useful for a variety of downstream tasks [21]. Reusing these weights shortens training time, reduces the risk of over-fitting, and lowers the amount of domain-specific data required. Accordingly, in the context of CNN, our framework instantiates three backbones—Inception, Xception, and InceptionResNet, as basic CNN models for image classification tasks, is well-founded, given their demonstrated effectiveness in various fields. The selection of these models pre-trained on the ImageNet dataset, which holds over a million images divided into 1,000 classes, has consistently demonstrated high performance in general image classification tasks [22]. Moreover, TL is not limited to CNN but extends to other architectures like LSTM networks, as seen in the forecasting of COVID-19 cases and deaths, where models trained on data from early-affected countries were used to predict outcomes in other regions, showcasing the method's applicability in time series analysis [23]. In each case, the bottom L_{frozen} convolutional blocks are kept fixed, while the remaining layers and the LSTM head are fine-tuned on IoTID20. The value of L_{frozen} is itself subject to optimization, enabling the system to balance feature reuse against task-specific adaptation.

The search space $\mathcal{H}$ is defined as follows: Let $\theta = \{\, \eta,\, p_{\text{drop}},\, \text{batch},\, L_{\text{frozen}},\, n_{\text{LSTM}},\, \text{epochs} \,\}$ collect the principal hyper-parameters: learning rate η, dropout probability p_{drop}, mini-batch size, number of frozen convolutional blocks, LSTM width, and epoch number. The Cartesian product of their admissible intervals defines a high-dimensional search space $\mathcal{H} \subset \mathbb{R}^6$. Optimizing CNN–LSTM models requires careful tuning of two hyper-parameter categories: model-design parameters (i.e., learning-rate schedule, proportion of frozen layers and dropout rate) that shape the network's architecture and regularization, and training parameters (i.e., epochs, batch size, and early-stopping patience) that control convergence and generalization. Because these settings strongly affect both accuracy and computational efficiency, automated search methods such as PSO, a population-based meta-heuristic algorithm that mimics the social behavior of birds or fish to converge on the best solution, are often employed to identify near-optimal configurations. To explore $\mathcal{H}$ efficiently, we adopt PSO, which performs a stochastic but directed search for the global optimum [24]. Each particle i maintains a position $\mathbf{x}_i(t) \in \mathcal{H}$ and velocity $\mathbf{v}_i(t)$. At every iteration t these are updated via (2) and (3).

$$\mathbf{v}_i(t+1)\& = w\,\mathbf{v}_i(t) + c_1 r_1\big(\mathbf{p}_i - \mathbf{x}_i(t)\big) + c_2 r_2\big(\mathbf{g} - \mathbf{x}_i(t)\big) \quad (2)$$

$$\mathbf{x}_i(t+1)\& = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (3)$$

Where w is the inertia weight, c_1 and c_2 are cognitive and social coefficients, $r_1, r_2 \sim \mathcal{U}(0,1)$, $\mathbf{p}_i$ is particle i's personal best, and $\mathbf{g}$ is the global best encountered so far.

During optimization, the fitness of each candidate $\theta \in \mathcal{H}$ is measured as indicated below: for each candidate θ, Algorithm 1 trains the CNN–LSTM core on the training subset of IoTID20 and returns the validation F1-score as the fitness value. After T iterations, PSO outputs $\theta^\star$, the hyper-parameter set that maximizes validation performance. These optimized parameters are then used to produce the final model weights $\Theta^\star$ deployed in the device and gateway-level IDS modules. The overall computational cost/complexity of a PSO routine is primarily dictated by three factors: the swarm size (N), the number of iterations (M), and the cost of a single fitness evaluation (F). For the standard PSO variant, the asymptotic running time can therefore be approximated as $O(N*M*F)$, where N represents the number of particles, M the total iterations, and F the complexity of evaluating the objective (i.e., fitness) function [25].

3. RESULTS AND DISCUSSION

This section presents the quantitative results obtained on the IoTID20 benchmark and evaluates the effectiveness of the proposed transfer-hybrid CNN–LSTM intrusion-detection framework. The obtained results are systematically compared with several state-of-the-art baseline methods to highlight performance differences in terms of accuracy, detection capability, and robustness. Furthermore, the observed performance gains are analyzed and discussed to emphasize their practical implications for real-world IoT-based smart-home security deployments.

3.1. Experimental configuration

All experiments were implemented in Python 3.9 with TensorFlow and Scikit-learn libraries. DL models were trained and evaluated on an ASUS ROG Strix G713QM laptop that incorporates an AMD Ryzen 9 5900HX processor running at a base clock of 3.3 GHz across sixteen logical cores, 16 gigabytes of system memory, and an NVIDIA GeForce RTX 3060 Laptop-class graphics processing unit with 6 of GDDR6 VRAM. The host operating system was the 64-bit edition of Windows 11. By exploiting TensorFlow's built-in GPU backend, the training pipeline made full and concurrent use of both the central and graphics processors, thereby accelerating forward and backward passes as well as hyper-parameter optimization. The proposed intrusion-detection framework was benchmarked on the IoTID20 dataset, introduced earlier in section 2.2. To obtain performance estimates that are both robust and unbiased, the evaluation adopted a five-fold cross-validation protocol. Because network-traffic corpora are characteristically imbalanced, legitimate flows greatly outnumber attack samples. Computational efficiency was quantified by recording end-to-end training time and inference time on the same hardware platform, which serves as a realistic proxy for an IoT-enabled smart-home edge device.

3.2. Hyper-parameter optimization results

To obtain the strongest variants of each backbone, all CNN–LSTM models were tuned with PSO on the training folds of IoTID20. The search spanned six principal hyper-parameters, number of epochs, batch size, early-stopping patience, learning rate, dropout probability, and LSTM width, plus the number of frozen convolutional layers specific to each backbone. Table 1 lists the explored ranges and the PSO-selected optimum for every model. PSO search converged within 35 iterations, after which performance metrics ceased to improve. These optimized settings were subsequently fixed for the five-fold cross-validation.

3.3. Performance analysis and evaluation

All models were assessed with five standard metrics: accuracy (Acc), precision (Prec), recall (Rec), F1-score (F_1), training time (Tr) in seconds, and inference (i.e., test) time per window in seconds (Ts). Three PSO-optimized TL variants, such as Xception–LSTM-PSO, Inception–LSTM-PSO, and InceptionResNet–LSTM-PSO, are compared against four reference IDS baselines widely cited in the literature. Table 2 summarizes the quantitative findings on the IoTID20 dataset.

All three proposed models achieve perfect scores across Acc, Prec, Rec, and F_1, demonstrating their ability to separate benign from malicious traffic and to discriminate among the four attack classes without

error. Among them, Xception-LSTM-PSO is the most efficient at run-time, requiring only 0.456 seconds per 228×76 window, followed by Inception-LSTM-PSO taking 1.017 seconds, and InceptionResNet-LSTM-PSO requiring 3.33 seconds. The performance gap to the strongest baseline CNN-SMOTE-PCA-BAT is modest in raw Acc, at around 0.03%, but sizeable in recall for the minority class, approximately 0.30%, which is critical for avoiding false negatives in safety-critical deployments. Relative to classification and regression trees (CART) and the reinforcement learning - deep Q-network (RL-DQN) agent, the proposed models deliver an absolute score of up to 99% and 99.44% in Acc, and eliminate false positives and negatives in those baselines. While the light gradient boosting machine (LightGBM) ensemble named optimized adaptive sliding window (OASW)-LightGBM-PSO achieves 99.96%, it remains 0.04% short of perfect classification, and its inference time 78 milliseconds is an order of magnitude faster than the two most lightweight CNN-LSTM variants. CNN-SMOTE-PCA-BAT pipeline attains 99.97% Acc but trails the proposed models by 0.26% in F_1 due to a lower recall for the port-scan class. Its training time (13.82 seconds) is longer than the Xception-based approach's, yet the latter achieves error-free performance. Among hybrid DL models, Xception-LSTM-PSO offers the best accuracy–latency trade-off, processing a 228×76 window in 0.456 seconds. This is well under the 10-second limit typically required for real-time protection on standard IoT devices. Although InceptionResNet-LSTM-PSO is three times slower (3.33 seconds), it may still be acceptable for gateway deployment where additional memory and compute are available. The results demonstrate that coupling TL with PSO-driven hyper-parameter tuning not only closes the performance gap to handcrafted feature pipelines but also pushes the detection ceiling to 100 % across all evaluation metrics, while maintaining practical inference latencies for embedded devices.

Table 1. CNN–LSTM hyper-parameter settings

Hyper-parameter	Search range	Backbone	Best value
Epochs	$[5,\ 21]$		15
Batch size	$\{32, 64, 128\}$		128
Early-stop patience	$\{2, 3, 4\}$		4
Learning rate	$(0.001,\ 0.006)$	All CNN-LSTM models	0.004
Dropout probability	$(0.30,\ 0.60)$		0.50
LSTM units	$\{64, 128, 256, 512\}$		256
Dense units	$\{64, 128, 256, 512\}$		256
Frozen layers	$[50, 131]$	Xception	111
	$[50, 150]$	Inception	124
	$[400, 550]$	InceptionResNet	466

Table 2. Model performance comparison on the IoTID20 dataset

Approach	Acc (%)	Prec (%)	Rec (%)	F_1 (%)	Tr (s)	Ts (s)
CART [26]	99.00	99.00	99.00	99.00	48.50	–
RL-DQN [27]	99.40	–	–	–	–	–
OASW-LightGBM-PSO [28]	99.92	99.93	99.98	99.96	–	0.0078
CNN-SMOTE-PCA-BAT [29]	99.97	99.79	99.70	99.74	13.82	–
Xception-LSTM-PSO	100.0	100.0	100.0	100.0	80.80	0.456
Inception-LSTM-PSO	100.0	100.0	100.0	100.0	224.17	1.017
InceptionResNet-LSTM-PSO	100.0	100.0	100.0	100.0	417.91	3.330

From a deployment perspective, Xception-LSTM-PSO provides the most favorable accuracy–latency trade-off for resource-constrained embedded IoT nodes. In contrast, the larger InceptionResNet-LSTM-PSO model, while incurring higher latency, is more suitable for gateway-level deployment where greater computational resources and memory are available. Notably, all evaluated variants operate well under the commonly cited 10-second real-time threshold for smart-home mitigation, thereby confirming their suitability for practical on-device inference.

4. CONCLUSION

Securing IoT-based smart home environments is a major challenge due to the growing number of interconnected devices, the vulnerability of these devices to cyber-attacks, and the potential impact on

user privacy and security. This paper presented a transfer-hybrid CNN–LSTM–based IDS tailored for IoT-based smart-home networks. By converting flow-level traffic into image-like tensors, exploiting pre-trained convolutional backbones, and optimizing all key hyper-parameters with PSO, the proposed framework achieves 100% accuracy, precision, recall, and F1-score on the IoTID20 benchmark while maintaining inference latency of less than 10 seconds on commodity hardware. Comparative experiments show a clear advantage over classical decision-tree and reinforcement-learning baselines and recent DL pipelines that rely on handcrafted feature engineering. The hierarchical design, lightweight device-level analysis complemented by gateway-level correlation, enables early containment of local compromises and robust detection of coordinated attacks, and covering internal and external threats. TL accelerates convergence and reduces the amount of annotated data needed, whereas PSO delivers resource-aware configurations without manual tuning. The findings demonstrate that the proposed transfer-hybrid architecture offers a practical, high-fidelity solution for securing next-generation smart homes against an increasingly complex threat landscape. The study assumes a balanced training set after oversampling; real smart-home traffic is significantly more skewed. Future work will investigate adaptive cost-sensitive loss functions and continual learning to counter concept drift. Moreover, adversarial robustness against gradient-based evasion remains an open challenge that will be addressed in subsequent research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT (MANDATORY)

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mouad Choukhairi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Ouail Choukhairi	✓	✓			✓	✓		✓		✓				
Youssef Fakhri		✓		✓	✓		✓			✓		✓	✓	
Ali Choukri				✓	✓					✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

This study did not involve human participants, and informed consent was therefore not required.

ETHICAL APPROVAL

This research did not involve human or animal subjects and did not require ethical approval.

DATA AVAILABILITY

The dataset underpinning this study is openly accessible in Google Sites at https://sites.google.com/view/iot-network-intrusion-dataset/home.

REFERENCES

[1] T. Magara and Y. Zhou, "Internet of things (IoT) of smart homes: privacy and security," *Journal of Electrical and Computer Engineering*, vol. 2024, pp. 1–17, Apr. 2024, doi: 10.1155/2024/7716956.

[2] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the internet of things," *Sensors*, vol. 19, no. 9, Apr. 2019, doi: 10.3390/s19091977.

[3] H. Alam and E. Tomai, "Security attacks and countermeasures in smart homes," *International Journal on Cybernetics & Informatics*, vol. 12, no. 2, pp. 109–119, Mar. 2023, doi: 10.5121/ijci.2023.120209.

[4] L. Santos, R. Gonçalves, C. Rabadão, and J. Martins, "A flow-based intrusion detection framework for internet of things networks," *Cluster Computing*, vol. 26, no. 1, pp. 37–57, Feb. 2023, doi: 10.1007/s10586-021-03238-y.

[5] S. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam, "Dependable intrusion detection system for IoT: a deep transfer learning based approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1006–1017, Jan. 2023, doi: 10.1109/TII.2022.3164770.

[6] M. M. Rahman, S. A. Shakil, and M. R. Mustakim, "A survey on intrusion detection system in IoT networks," *Cyber Security and Applications*, vol. 3, Dec. 2025, doi: 10.1016/j.csa.2024.100082.

[7] M. Anwer, S. M. Khan, M. U. Farooq, and . Waseemullah, "Attack detection in IoT using machine learning," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7273–7278, Jun. 2021, doi: 10.48084/etasr.4202.

[8] H. Liao *et al*., "A survey of deep learning technologies for intrusion detection in internet of things," *IEEE Access*, vol. 12, pp. 4745–4761, 2024, doi: 10.1109/ACCESS.2023.3349287.

[9] M. Afroz, E. Nyakwende, and B. Goswami, "Intrusion detection in smart home environments: a machine learning approach," *Transportation Research Procedia*, vol. 84, pp. 243–250, 2025, doi: 10.1016/j.trpro.2025.03.069.

[10] N. Karmous, Y. Ben Dhiab, M. O.-Elhassen Aoueileyine, N. Youssef, R. Bouallegue, and A. Yazidi, "Deep learning approaches for protecting IoT devices in smart homes from MitM attacks," *Frontiers in Computer Science*, vol. 6, Oct. 2024, doi: 10.3389/fcomp.2024.1477501.

[11] P. Sinha, D. Sahu, S. Prakash, T. Yang, R. S. Rathore, and V. K. Pandey, "A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning," *Scientific Reports*, vol. 15, no. 1, Mar. 2025, doi: 10.1038/s41598-025-94500-5.

[12] A. Nazir *et al*., "A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem," *Ain Shams Engineering Journal*, vol. 15, no. 7, Jul. 2024, doi: 10.1016/j.asej.2024.102777.

[13] A. Deshmukh and K. Ravulakollu, "An efficient CNN-based intrusion detection system for IoT: use case towards cybersecurity," *Technologies*, vol. 12, no. 10, Oct. 2024, doi: 10.3390/technologies12100203.

[14] S. Abdelhamid, I. Hegazy, M. Aref, and M. Roushdy, "Attention-driven transfer learning model for improved IoT intrusion detection," *Big Data and Cognitive Computing*, vol. 8, no. 9, Sep. 2024, doi: 10.3390/bdcc8090116.

[15] A. Qaddos, M. U. Yaseen, A. S. Al-Shamayleh, M. Imran, A. Akhunzada, and S. Z. Alharthi, "A novel intrusion detection framework for optimizing IoT security," *Scientific Reports*, vol. 14, no. 1, Sep. 2024, doi: 10.1038/s41598-024-72049-z.

[16] B. Isong, O. Kgote, and A. A.-Mahfouz, "Insights into modern intrusion detection strategies for internet of things ecosystems," *Electronics*, vol. 13, no. 12, Jun. 2024, doi: 10.3390/electronics13122370.

[17] P. Aravamudhan and K. T, "A novel adaptive network intrusion detection system for internet of things," *PLoS ONE*, vol. 18, no. 4, Apr. 2023, doi: 10.1371/journal.pone.0283725.

[18] S. Gallón, J.-M. Loubes, and E. Maza, "Statistical properties of the quantile normalization method for density curve alignment," *Mathematical Biosciences*, vol. 242, no. 2, pp. 129–142, Apr. 2013, doi: 10.1016/j.mbs.2012.12.007.

[19] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114–119, Jun. 2019, doi: 10.1109/MCOM.2019.1800155.

[20] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, Sep. 2022, doi: 10.1109/ACCESS.2022.3206425.

[21] H.-C. Shin *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016, doi: 10.1109/TMI.2016.2528162.

[22] Z. Zhao, "A comparative study of large-scale and lightweight convolutional neural networks for ImageNet classification," *Applied and Computational Engineering*, vol. 47, no. 1, pp. 101–110, Mar. 2024, doi: 10.54254/2755-2721/47/20241236.

[23] Y. Gautam, "Transfer Learning for COVID-19 cases and deaths forecast using LSTM network," *ISA Transactions*, vol. 124, pp. 41–56, May 2022, doi: 10.1016/j.isatra.2020.12.057.

[24] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.

[25] A. Kaveh, "Particle swarm optimization," in *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Cham, Switzerland: Springer International Publishing, 2021, pp. 13–46, doi: 10.1007/978-3-030-59392-6_2.

[26] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet of Things*, vol. 3, no. 1, May 2023, doi: 10.1007/s43926-023-00034-5.

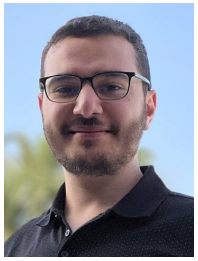
[27] T. V. Ramana, M. Thirunavukkarasan, A. S. Mohammed, G. G. Devarajan, and S. M. Nagarajan, "Ambient intelligence approach: Internet of things based decision performance analysis for intrusion detection," *Computer Communications*, vol. 195, pp. 315–322, Nov. 2022, doi: 10.1016/j.comcom.2022.09.007.

[28] L. Yang and A. Shami, "A lightweight concept drift detection and adaptation framework for IoT data streams," *IEEE Internet of Things Magazine*, vol. 4, no. 2, pp. 96–101, Jun. 2021, doi: 10.1109/IOTM.0001.2100012.

[29] H. Karamollaoğlu, İ. A. Doğru, and İ. Yücedağ, "An efficient deep learningbased intrusion detection system for internet of things networks with hybrid feature reduction and data balancing techniques," *Information Technology and Control*, vol. 53, no. 1, pp. 243–261, Mar. 2024, doi: 10.5755/j01.itc.53.1.34933.

BIOGRAPHIES OF AUTHORS

Mouad Choukhairi received his bachelor's degree (B.Sc.) in 2018 in Mathematics and Computer Science and his master's degree in Big Data and Cloud Computing in 2020 from the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. He is currently a Ph.D. student in the Computer Science Research Laboratory (LaRI) in cybersecurity, IDS, big data, ML, DL, and IoT. He can be contacted at email: mouad.choukhairi@uit.ac.ma.



Ouail Choukhairi received his bachelor's degree (B.Sc.) in 2020 in Mathematics and Computer Science and his master's degree in Big Data and Cloud Computing in 2022 from Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. He is currently a Ph.D. student in the Computer Science Research Laboratory (LaRI) in the field of artificial intelligence, semantic segmentation, ML, and DL. He can be contacted at email: ouail.choukhairi@uit.ac.ma.

Youssef Fakhri obtained a bachelor of Science degree in Electronics and a diploma of Advanced Scientific Studies (DESA) in Computer Science and Telecommunications from the Faculty of Sciences of Rabat (Mohammed V University - Agdal, Rabat, Morocco), in 2001 and 2003 respectively. He obtained a Ph.D. thesis on November 17, 2007, from the University Mohammed V - Agdal, Rabat, Morocco in collaboration with the Polytechnic University of Catalonia (UPC), Spain. His research topics are information theory, signal processing, WSN, and routing protocols. He is now a professor of Higher Education in Computer Science at the Faculty of Sciences of Kenitra. He is the head of the Research Laboratory in Computer Science. He can be contacted at email: fakhri@uit.ac.ma.

Ali Choukri is an assistant professor at the National Academy of Applied Sciences. He completed his master's degree in Computer Science and Telecommunications in 2008 at Ibn Tofail University in Kenitra, Morocco. He holds a Ph.D. from the School of Computer Science and Systems Analysis (ENSIAS) and a degree from the Higher Normal School of Technical Teaching (ENSET). He is currently part of the Management and Intelligent Systems (MIS) team in the Signals, Systems, and Embedded Intelligence Laboratory (SIME). His research focuses on mobile intelligent ad hoc communication systems and wireless sensor networks. He is also interested in areas such as ubiquitous computing, the internet of things (IoT), quality of service (QoS) routing, mathematical modeling, game theory, and optimization. He can be contacted at email: ali.choukri@uit.ac.ma.