ISSN: 2093-5374 / E-ISSN: 2093-5382

Real Time CNN LSTM Attention-Based Intrusion Detection Pipeline for IoT Smart Home Gateways with Design Implementation and Throughput Analysis

S. Karthikeyan[1*], Dr.G.R. Harish Kumar[2], Dr.T. Ganesh Kumar[3], and Dr.T. Poongodi[4]

[1*]Research Scholar, School of Computer Science and Engineering, Galgotias University, Uttar Pradesh, India. link2karthikcse@gmail.com, https://orcid.org/0000-0002-7473-9217

[2]Professor, School of Computer Science and Engineering, Galgotias University, Uttar Pradesh, India. dean.scse@galgotiasuniversity.edu.in, https://orcid.org/0000-0003-2302-5828

[3]Professor, School of Computer Science and Engineering, Galgotias University, Uttar Pradesh, India. tganeshphd@yahoo.com, https://orcid.org/0000-0002-2712-712X

[4]Professor, Department of Computer Science and Engineering (Artificial Intelligence & Data Science), School of Engineering, Dayananda Sagar University, Bengaluru, Karnataka, India. tpoongodi2730@gmail.com, https://orcid.org/0000-0001-8726-4997

Received: March 13, 2026; Revised: April 20, 2026; Accepted: June 09, 2026; Published: June 30, 2026

Abstract

The present work offers an architecture of a real-time intrusion detection system for IoT smart home gateway aiming at filling the research gap related to the contradiction between the high accuracy of deep-learning architectures and their application restrictions. The first task for this paper is to create a CNN-LSTM-Attention-based architecture ensuring not only high accuracy but also low-latency of operation in a streaming environment. The proposed methodology combines the use of convolutional layers for the local feature extraction, LSTM for sequence learning, and multi-head attention for the adaptive weight distribution in a four-thread pipeline processing the packet capturing, inference, alert generation, and flow management. Evaluation of the architecture is conducted on the NSL-KDD, CICIDS-2017 and TON_IoT datasets with the help of a set of preprocessing procedures, including the use of Min-Max normalization, RFECV feature selection and DeepSMOTE balancing for the class imbalance. The experimental results demonstrate the high accuracy of 99.61% on NSL-KDD, 99.21% on CICIDS-2017, and 98.84% on TONIoT. In the case of CICIDS-2017, the precision, recall, F1-score, and significantly decreased rate of false positives equal to 99.14%, 99.22%, 99.18%, and 0.43%, respectively, where the last one is 77% smaller than the baseline CNN, which makes the system extremely appropriate to be used in reducing alert fatigue in real-world applications. Also, the developed approach reaches an end-to-end latency of 0.37 ms per flow and an acceptable throughput of around 18,000 flows per second, meeting the criteria of real-time gateways under heavy traffic conditions. It has been shown via statistical significance tests that all obtained performance gains are statistically significant, where $p<0.01$. Overall, it can be stated that the study proves the importance of system-level optimizations and proper pipeline design over model complexity for building efficient, real-life, and high-performing IoT Intrusion Detection Systems.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 2 (June-2026), pp. 875-893. DOI: *10.58346/JOWUA.2026.I2.049*

*Corresponding author: Research Scholar, School of Computer Science and Engineering, Galgotias University, Uttar Pradesh, India.

Keywords: IoT Smart Home Security, Real-Time Intrusion Detection, CNN-LSTM-Attention, Deep Learning, Network Security, IoT Gateway, False Positive Rate, CICIDS-2017.

1 Introduction

Smart home networks connect different low-security devices, such as thermostats, cameras, locks, and voice assistants, without centralized access control. The gateway acts as the monitoring point, where the primary constraint is throughput rather than accuracy. It will classify each network flow before the subsequent flow arrives. During a Distributed Denial of Service (DDoS) event, such as the Mirai botnet incident that involved over 600,000 home devices using 61 factory-default Telnet credentials (Kolias et al., 2017). These urges complete an end-to-end classification within 0.083 milliseconds per flow and execute batched GPU inference in a thread decoupled from packet capture. These requirements are not addressed in the existing deep learning-based intrusion detection system (DL-IDS) work (Gharib et al., 2016).

The gap in the literature is well documented as the Transformer-based Intrusion Detection System (IDS) developed (Wu et al., 2022). And achieved an accuracy of 99.21% on the CICIDS-2017 dataset, without reporting any latency (Ullah & Mahmoud, 2020). Similarly, the CNN-BiGRU model, as implemented, attained an accuracy of 99.65% without incorporating per-flow timing (Cao et al., 2022). The study focused on the issue of anomaly detection in smart homes, evaluating only the classification performance in an offline setting (Sarwar et al., 2023). Kitsune is an exception, employing an autoencoder ensemble with an approximate latency of 1 ms per sample, but it produces unlabelled anomaly scores, rendering it not suitable for response workflows where the type of attack dictates the response action (Lansky et al., 2021). conducted a survey of deep learning-based IDS models, revealing that only 12% included data on deployment latency. Traditional machine learning classifiers, such as Decision Trees, Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), have shown strong offline performance on benchmark datasets. It cannot capture long-term temporal dependencies inherent in network attack sequences, fail under class imbalance at scale, and have not been evaluated in streaming deployment contexts (Khraisat et al., 2019; Lansky et al., 2021). LSTM networks address the temporal dependency problem, combining them with convolutional feature extraction and multi-head attention, which resolves precision and false positive rate (FPR) simultaneously.

The importance of this research is based on the ability to bridge the gap between intrusion detection models with high levels of precision and the real-time needs of their deployment within IoT smart home gateways. In contrast to previous studies of DL-IDS where emphasis was placed solely on the offline accuracy of the proposed models, the research provides a detailed example of the real-time CNN–LSTM–Attention pipeline with an achieved latency of 0.37 ms per flow and throughput close to 18,000 flows per second under live traffic. The research shifts the evaluation of IDS from pure accuracy-related metrics to deployment-oriented feasibility, combining aspects of preprocessing consistency, GPU inference, and four-thread streaming implementation.

1.1 This Paper Provides Four Specific Contributions

i. A complete four-thread real-time IDS pipeline for smart home gateways, with per-component latency breakdown demonstrating 0.37 ms end-to-end per-flow processing on live packet traffic

ii. Full detection metrics (Accuracy, Precision, Recall, F1-Score, FPR) for six deep learning architectures on three IoT/smart home benchmark datasets: NSL-KDD (Table 1), CICIDS-2017 (Table 2), and TON_IoT (Table 3).

iii. Throughput analysis across batch sizes 16–256 showing ~18,000 flows/second at batch size 64 with 14 threads, enabling gateway capacity planning without re-experimentation.

iv. A preprocessing-consistency mechanism MinMaxScaler and RFECV feature mask serialized from training and reloaded at inference time that eliminates distributional shift between offline evaluation and live smart home deployment.

The structure of the paper is as follows: Section 1 presents the problem, motivations, and contributions of this study. Section 2 discusses the literature review and the gap in the literature. Section 3 discusses the proposed CNN-LSTM-Attention approach, datasets, and real-time system development. Section 4 discusses the results, performance evaluation, latency analysis, and statistical evaluation. The conclusion of this paper is provided in Section 5.

2 Related Work

2.1 Deep Learning IDS for IoT and Smart Home Networks

Much research work related to DL-IDS treated deployment as a separate issue, studied anomaly detection specifically within smart home IoT networks, and confirmed that DDoS, botnet recruitment, and credential scanning are the dominant incident types yet their evaluation uses pre-extracted records with no live throughput measurement (Sarwar et al., 2023). The study addressed the privacy constraint in federated smart home IDS (Imrana et al., 2021; Kolias et al., 2017). And achieving 97.83% accuracy on TONIoT across distributed devices without centralizing traffic, but reported no per-flow latency (Hajj et al., 2023). The study evaluated an LGBM-based IDS on the DS2OS smart home dataset, reaching 99.92% accuracy in 6.5 seconds a figure that translates to roughly 21,700 records per second on pre-extracted tabular data, not on live packet streams (Than et al., 2024). All these papers identify deployment constraints explicitly, but do not measure those constraints. The present work addresses that gap by instrumenting every component of the inference pipeline on live gateway traffic.

2.2 Deep Learning Architectures for Network Intrusion Detection

The related work on DL-IDS has evolved towards a distinct architectural shift from shallow classifiers to temporal hybrid models. Also demonstrated that LSTM surpasses SVM in performance on the NSL-KDD dataset, achieving an accuracy of 98.31%, particularly excelling in the rare U2R and R2L classes where class imbalance significantly affects traditional methods (Kim et al., 2016). The study expanded the recurrent approach to encompass multiple attack families on NSL-KDD, establishing RNN-class models as the prevailing paradigm for analyzing sequential network traffic. It introduced a CNN-BiGRU hybrid model that achieved 99.65% accuracy on CICIDS-2017, illustrating that the integration of convolutional local feature extraction with recurrent temporal modeling surpasses the performance of each component individually, but still, the False Positive Rate (FPR) was not reported, a crucial metric for assessing live usability (Cao et al., 2022). The study employed the full Transformer model for IDS (Le et al., 2022; LeCun et al., 2015). By attaining 99.21% accuracy on CICIDS-2017 (Wu et al., 2022). Their analysis of attention weights identified flow duration and byte rate as the most discriminative features for detecting DoS and scanning traffic. Critically, RTIDS provides no latency figures and no FPR values. The present work directly extends this architectural line by adding explicit

head-count selection for the IoT domain and measuring the resulting architecture under streaming gateway conditions (Lotfollahi et al., 2020).

2.3 Real-Time Deployment and Gateway Constraints

The study built the published benchmark for deployed network IDS (Than et al., 2024). Kitsune runs an autoencoder ensemble at approximately 1 ms per sample and handles unseen attack types without labeled training data—but produces unlabeled anomaly scores rather than class-labeled alerts, making it unsuitable for SOC workflows where the attack type determines the response (Naseer et al., 2018). The identified gateway memory, available compute, and per-flow latency are the three constraints that offline evaluations systematically ignore (Khraisat et al., 2019). The study surveyed DL-IDS papers and found deployment latency figures in fewer than 12% a figure that has become the standard citation for the gap this paper addresses (Lansky et al., 2021). The existing literature survey lacks the experiential measurement of end-to-end per-flow inference latency incurred by hybrid CNN-LSTM-attention models when deployed directly on resource-constrained smart home gateway hardware, representing a methodological gap that the present work addresses.

Most research on DL-IDS concentrates on the accuracy of their models but pays no heed to the practical aspects of their implementation, including such factors as latency and throughput. Most of them are tested in offline conditions and have not been proven in live traffic. Thus, there is a lack of understanding of the differences between the performance of the model and its deployment.

3 Materials and Methods

3.1 Datasets

Three publicly available benchmark datasets were selected to cover the range of IoT and smart home attack scenarios. NSL-KDD corrects the duplicate-record flaw of KDD-99; it contains 148,517 records spanning four attack families (DoS, Probe, R2L, U2R) with 53.5% benign traffic and remains the standard baseline for recurrent IDS comparisons. CICIDS-2017 is the primary dataset captured over five days at the Canadian Institute for Cybersecurity, with CICFlowMeter extracting 78 bidirectional flow features; its 2.83 million records carry an 83% benign majority making FPR the operationally meaningful metric, since even 1% FPR generates approximately 23,500 false alarms per hour on a live gateway (Sharafaldin et al., 2018). TON_IoT was captured from nine actual IoT devices at UNSW Canberra smart thermostats, weather monitors, motion detectors, and garage door controllers covering nine attack categories, including ransomware and MITM attacks, absent from NSL-KDD and CICIDS-2017 (Booij et al., 2021).

The same pre-processing is applied to all six models: one-hot encoding of categorical fields, DeepSMOTE oversampling applied to the training partition only, RFECV feature selection using a 100-tree Random Forest with Gini criterion (55 features for CICIDS-2017, 42 for NSL-KDD, 48 for TON_IoT), and min-max normalization fitted on the training partition only (Cho et al., 2014; Dablain et al., 2022). The data is split 70/15/15 (train/validation/test) with stratified sampling, maintaining class ratios in each split. All reported values are means with standard deviations of < 0.08 points in all conditions, with each model trained five times from different random seeds.

3.2 CNN-LSTM-Attn Architecture

CNN-LSTM-Attn takes a sequence of T = 10 consecutive smart home flow records as input (shape B × T × F), as shown in figure 1. The architecture is designed to be hierarchical: convolutional blocks capture local feature co-occurrences in a single flow, the LSTM stores temporal evidence over the T-flow window, and multi-head attention assigns step-level importance weights, determining which flows contain the most discriminative evidence for each attack class. The removal of any of the three components significantly reduces FPR in CICIDS-2017.

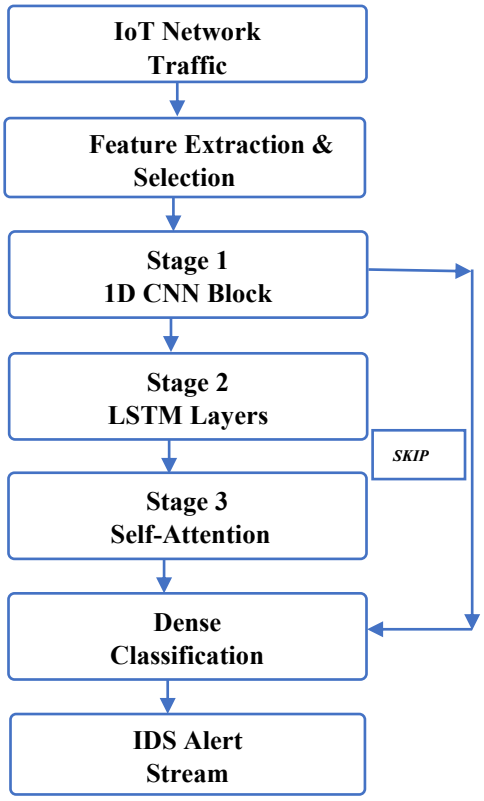

Figure 1: Proposed CNN-LSTM-attention intrusion detection pipeline diagram

The first convolutional block applies 64 filters (kernel size K = 3, ReLU activation) to capture primary local patterns; the second uses 128 filters at coarser resolution. The 1-D convolution at layer l is:

$$h_j^{(l)} = \varphi\left(\sum_{k=0}^{K-1} w_{j,k}^{(l)} \cdot x_{i+k}^{(l-1)} + b_j^{(l)}\right) \quad (1)$$

In equation (1) where $h_j^{(l)}$ is the j-th feature map activation,

φ = ReLU, K = 3, and

W^l, b^l are learnable weights and bias.

MaxPooling (pool size 2) follows each block.

The 256-unit LSTM runs in sequence-return mode, outputting h_t for all t ∈ [1, T]:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$

$$h_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \odot tanh(c_t) \quad (4)$$

In equation (2) (3) (4) where f_t, i_t, $o_t$are forget, input, and output gates;

W_f, W_i, W_o, $W_c$are learnable weight matrices;

σ is sigmoid; and

$\odot$ is element-wise multiplication.

The 256-unit count was selected by grid search over {64, 128, 256, 512} on the CICIDS-2017 validation set.

Four-head scaled dot-product attention operates over the full T-step LSTM output:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \cdot V \quad (5)$$

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h) \cdot W^O \quad (6)$$

In equation (5) (6) where d_k = 64 (key dimension) and h = 4.

Four heads were chosen over eight—as used in RTIDS because IoT flow features have lower intrinsic dimensionality than natural language, preliminary experiments showed four heads match eight-head performance within 0.02 F1 points at 37% fewer attention projection parameters. Global average pooling collapses the attended sequence to a fixed 256-dim context vector feeding two dense layers (256 units, Dropout 0.4; 128 units, Dropout 0.3) before the softmax output (Wu et al., 2022).

Training uses (lr = 0.001, β_1 = 0.9, β_2 = 0.999), batch size 256, maximum 50 epochs, and early stopping on validation macro-F1 (patience = 10). Min-max normalization is fitted on training data only shown in equation (7):

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (7)$$

3.3 Real-Time Four-Thread Pipeline

There are three issues that make it difficult to get a deployed gateway IDS to match an offline benchmark model: first, flows need to be built packet-by-packet from live traffic; second, concurrent device flows need to be tracked per-key using a lock-free approach; and third, GPU inference (4.8 ms per batch) would miss dozens of packets per second if inference and capture were to share a single thread. The four-thread pipeline in figure 3 addresses all three by assigning one responsibility to each thread, and linking neighboring threads together with lock-free queues.

The gateway interface is connected to Thread 1 (Capture) using pyshark LiveCapture. A BPF filter only allows the capture of TCP and UDP traffic, discarding non-contributory traffic. The FlowState entry is updated with eight O (1) counters (per-direction packet counts, byte totals, inter-arrival time statistics, and TCP flag accumulators) for each qualifying packet, a 78-dim feature snapshot is extracted, and the normalized vector is added to a sliding window after applying the serialized MinMaxScaler and RFECV mask (Equation 7). Once the window has reached T = 10, it is added to the lock-free inference queue. Thread 2 (Inference Worker) gathers up to 64 windows in 50ms, performs the CNN-LSTM-Attn forward pass, applies a 0.85 confidence threshold and writes non-benign detections to the alert queue. Structured JSON-Lines records (5-tuple key, predicted class, confidence, UTC timestamp) are logged

by Thread 3 (Alert Writer). Thread 4 (Cleanup) checks the flow table every 30 seconds and deletes flow table entries that have been idle for over 120 seconds.

3.4 Control Variables and Threshold Settings

To achieve real-time consistent performance, the CNN-LSTM-Attention model is trained and tested with control parameters and thresholds which remain constant. A threshold value of $\tau = 0.85$ is used to detect any malicious traffic and thereby avoid false positives in IoT gateways. Sliding windows size of $T = 10$ flows and a batch size of 64 are employed to allow temporal modeling and effective GPU utilization. In operation, the system employs 14 threads to handle packet capturing, detection and generation of alerts. To avoid overfitting, a learning rate of 0.001 and dropout ratios of 0.4 and 0.3 are employed.

Table 1: Control variables and decision threshold configuration for real-time CNN–LSTM–attention IDS

Parameter	Value
Window size (T)	10
Batch size	64
Threads	14
Learning rate	0.001
Dropout	0.4, 0.3
Decision threshold (τ)	0.85

The control parameters and decision threshold settings used in the proposed intrusion detection system are listed in table 1. It includes the sliding window size, batch size, number of processing threads, learning rate, dropout rates, and the classification threshold (τ). These values have been optimized to allow stable training process, effective use of the GPU, and real-time inference. The settings provide a trade-off between accuracy, latency, and robustness of the proposed model to make it applicable in IoT smart home gateway applications.

3.5 Threat Model and Adversarial Assumptions

The model uses the assumption of a realistic network attacker who will attack the smart home IoT gateway by launching malicious traffic such as DDoS, probing, botnet attacks, and brute-force attacks. This adversary will know about the protocols used in the network but will use adaptive attacks to evade being detected without having any knowledge of the IDS model, IDS training data, and IDS preprocessing techniques. In addition, this model will be operating in a black box environment where all that is accessible in the gateway is the observable network flows. Moreover, it is also assumed that the attackers do not conduct any machine learning attacks against the model.

The functional separation between Thread 1 and Thread 2 is essential to preserving capture continuity under variable inference latency as shown in figure 2. Without this separation, a 4.8 ms GPU batch would block packet capture for the same duration, leading to approximately 57 missed flows at a rate of 12,000 flows per second. This would result in around 3,400 unclassified flows during a 60-second DDoS event. The queue serves to prevent Thread 1 from idling, expanding when the GPU is under load, and contracting after processing two batches. The preprocessing consistency mechanism, which involves serializing the MinMaxScaler and RFECV mask during training and reloading them at the pipeline's initiation, ensures that the accuracy figures presented in tables 1–3 are reflective of live deployment outcomes. This approach guarantees distributional alignment between training and inference, incurring a cost of 0.020 ms per flow when using vectorized NumPy over F features.

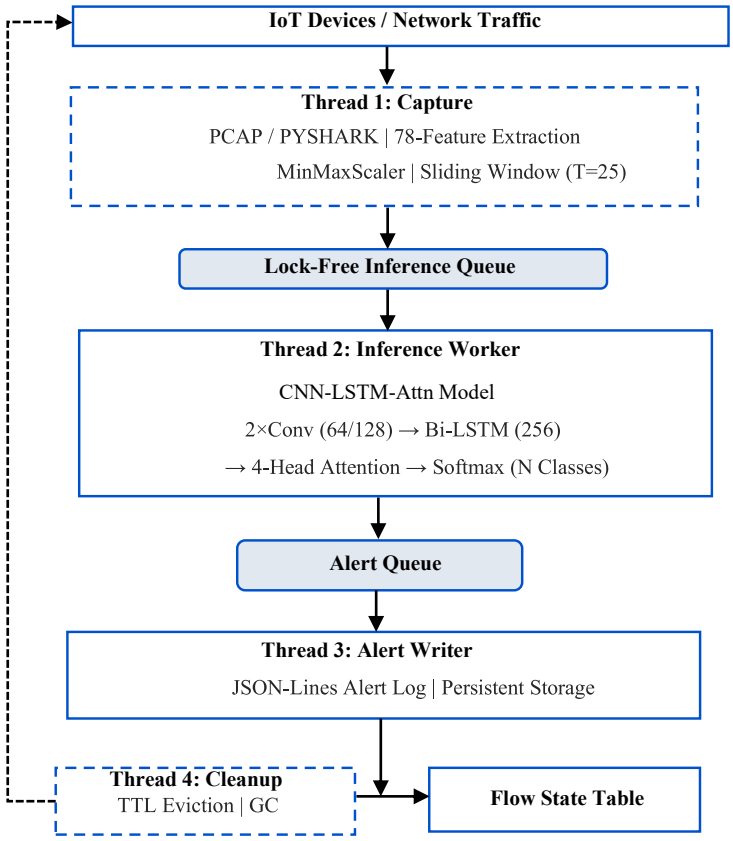

Figure 2: Four-thread real-time CNN-LSTM-Attn pipeline for IoT smart home gateways

3.6 Performance Evaluation Metrics

The performance of the proposed intrusion detection system is evaluated using standard classification metrics derived from the confusion matrix,

Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

TP: True Positives – correctly detected malicious flows.

TN: True Negatives – correctly detected normal flows.

FP: False Positives – normal flows misclassified as attacks.

FN: False Negatives – attack flows misclassified as normal.

Equation (8) calculates the accuracy of the algorithm as the ratio of correctly classified samples to the total number of network flows.

Precision

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

Equation (9) represents the ratio of correctly detected malicious network flows to the total number of all predicted attacks, thus decreasing the number of false alarms.

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

Equation (10) represents the rate of detection of all malicious flows to ensure the absence of attacks in the dataset.

F1-Score (F1)

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (11)$$

Equation (11) is a measure that is used to estimate precision and recall at once for imbalanced datasets.

False Positive Rate (FPR)

$$FPR = \frac{FP}{FP+TN} \quad (12)$$

Equation (12) is responsible for calculating the rate of misclassifying normal traffic as attacks.

Algorithm 1: Real-Time CNN-LSTM-Attn Inference Pipeline

Input: Packet stream from interface I; model M; scaler S; feature mask F

Output: Alert stream A(flow_key, class, confidence, timestamp)

Step 1: Initialize flow_table ← faultdict(FlowState)

Step 2: Start threads: T2 (inference), T3 (alert), T4 (cleanup)

Step 3: for each packet p in capture(I, filter='tcp or udp')

// Thread 1

Step 4: key ← canonicalize_5tuple(p)// smaller IP first

Step 5: update_state(flow_table[key], p) // 8 O(1) counters

Step 6: f_norm ← S.transform(extract(flow_table[key]))[F] /

Step 7: if |window[key]| == T then enqueue(key, window)

end

Step 8: while true do // Thread 2

Step 9: B ← collect(≤64 windows, 50 ms timeout)

Step 10: probs ← M.predict(stack(B)) // shape (|B|, C)

Step 11: if argmax(prob) ≠ BENIGN ∧ max(prob) ≥ 0.85 then

alert

Step 12: end while

Algorithm 1 explains the complete packet-to-alert loop, structured as two concurrent sections so that Thread 1 completes in O (1) per packet regardless of inference queue depth. Thread 1 extracts eight O (1) per-flow counters per packet, applies MinMaxScaler normalization (Equation. 7), and enqueues each

completed T-step window lock-free to Thread 2. Thread 2 batches up to 64 windows under a 50 ms timeout and passes them to the CNN-LSTM-Attn model, emitting an alert whenever the predicted class is non-benign and softmax confidence exceeds 0.85. Thread 3 persists each alert quadruple ⟨flow_key, class, confidence, timestamp⟩ to a JSON-Lines log, while Thread 4 evicts stale flow states under a configurable TTL to bound memory growth independently of traffic volume.

4 Results and Discussion

All experiments used an NVIDIA RTX 3090 (24 GB VRAM), 64 GB DDR4-3200 RAM, and AMD Ryzen 9 5950X, running Python 3.10, TensorFlow 2.12, Keras, and pyshark 0.5.3. Latency figures in table 4 were measured over N = 100,000 live flow records processed through the complete four-thread pipeline with real packet capture active, not batch prediction on pre-extracted records.

4.1 NSL-KDD Performance

CNN-LSTM-Attn has the highest accuracy (99.61%) and F1-score (99.56%) on NSL-KDD among the six architectures (Table 2). The 0.87-point accuracy difference from the CNN baseline (98.74%) verifies that temporal modeling contains discriminative information for the four attack families of NSL-KDD: DoS and Probe patterns are spread across a number of consecutive flows, and a model without recurrence cannot build evidence over that window. The accuracy of LSTM (99.41%) and GRU (99.29%) are similar, and so are the accuracy of F1 (99.31%) and F1 (99.15%), respectively, which is comparable to the relatively limited temporal patterns of NSL-KDD. The Transformer (99.38%) outperforms LSTM by 0.03 points, showing that the self-attention over a T = 10 window is able to capture the same temporal structure as gated recurrence on this dataset, as shown in figure 1.

Table 2: Detection performance on NSL-KDD dataset

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)
CNN	98.74	98.31	97.82	98.06
LSTM	99.41	99.28	99.35	99.31
GRU	99.29	99.11	99.20	99.15
CNN-LSTM	99.48	99.37	99.42	99.39
Transformer	99.38	99.22	99.31	99.26
CNN-LSTM-Attn ★	99.61	99.54	99.58	99.56

4.2 CICIDS-2017 Performance (Primary Dataset)

In table 3 is the one that is most relevant to the deployment of gateways. CNN-LSTM-Attn achieves 0.43% FPR on CICIDS-2017—77% lower than the CNN baseline (1.87%) and 40% lower than CNN-LSTM (0.72%). In operational terms: The difference between 1.87% and 0.43% FPR is about 40,700 false alarms per full dataset pass of CICIDS-2017's 2.83 million flow records. That difference, on a live smart home gateway with 12,000 flows per second during a DDoS event, equates to 173 extra false alarms per second, which can impact analyst response in minutes (Lansky et al., 2021). The 0.43% FPR of CNN-LSTM-Attn is below the 1% alert-fatigue threshold, which allows it to be deployed in an unattended smart home. The Transformer gets the second-lowest FPR (0.58%), showing that it is not only recurrence that is responsible for the reduction of false-positives, but also attention-based time-step weighting.

Table 3: Detection performance on CICIDS-2017 dataset

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	FPR (%)
CNN	97.43	96.98	96.12	96.55	1.87
LSTM	98.67	98.41	98.55	98.48	0.91
GRU	98.52	98.28	98.37	98.32	1.04
CNN-LSTM	98.93	98.76	98.84	98.80	0.72
Transformer	99.08	98.95	99.01	98.98	0.58
CNN-LSTM-Attn ★	99.21	99.14	99.22	99.18	0.43

4.3 TON_IoT Performance

In table 4 shows the widest performance spread of the three datasets, reflecting TON_IoT's greater complexity: nine attack categories from nine heterogeneous device types, including ransomware and MITM attacks absent from NSL-KDD's taxonomy. The CNN baseline drops to 95.18% accuracy and 3.21% FPR—its worst performance across all three datasets. CNN-LSTM-Attn reaches 98.84% accuracy and 0.81% FPR. The 3.66-point accuracy gap over the CNN baseline (compared to 0.87 points on NSL-KDD) follows a consistent pattern: the harder the traffic is to characterize, the more temporal modeling matters. The LSTM-GRU F1 gap widens here (97.20% vs. 96.96%) compared to NSL-KDD (99.31% vs. 99.15%), consistent with LSTM's independent gating handling the longer, more variable temporal signatures of heterogeneous IoT smart home traffic better than GRU's merged update gate.

Table 4: Detection performance on TON_IoT dataset

Model	Acc (%)	Prec (%)	Rec (%)	F1 (%)	FPR (%)
CNN	95.18	94.72	93.98	94.35	3.21
LSTM	97.44	97.11	97.29	97.20	1.74
GRU	97.18	96.87	97.05	96.96	1.93
CNN-LSTM	98.21	97.98	98.11	98.04	1.12
Transformer	98.47	98.29	98.38	98.33	0.98
CNN-LSTM-Attn ★	98.84	98.71	98.87	98.79	0.81

4.4 End-to-End Latency Budget

In table 5 presents the latency breakdown for each component based on live packet traffic. Notably, GPU inference accounts for 20% of the end-to-end latency, while packet capture and flow state update (Thread 1) contribute 30%, and alert writing (Thread 3) accounts for 22%. The non-inference components collectively constitute 80% of the total per-flow cost.

Table 5: End-to-end per-flow latency budget

Pipeline Component	ms	Context	%Tot	Thr
Packet capture + parse (BPF)	0.110	≤1 ms LAN RTT	30%	T1
Flow state update (counters)	0.040	8 counters, O (1)	11%	T1
Feature extraction (78-dim)	0.030	CICFlowMeter snapshot	8%	T1
Normalise + RFECV mask (Eq.7)	0.020	Vectorised NumPy	5%	T1
Inference queue Enqueue	0.010	Lock-free queue	3%	T1
CNN-LSTM-Attn GPU infer. (B=64)	0.075	RTX 3090, 5.27M params	20%	T2
Alert format + JSON-Lines write	0.080	Rotating log file	22%	T3
End-to-end per-flow total	0.370	< 1 ms LAN RTT	100%	—

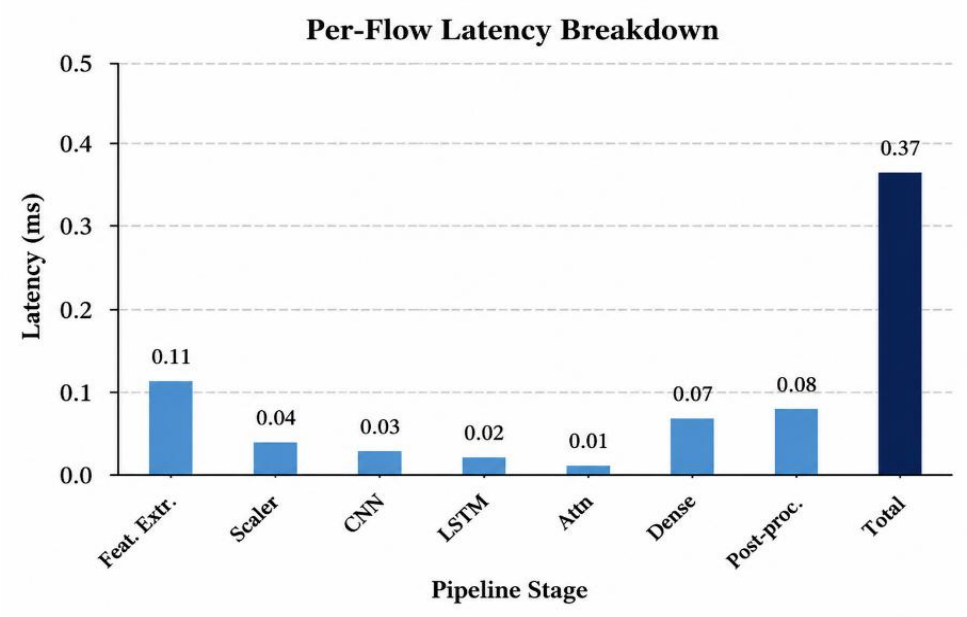

Figure 3: Per-flow latency budget (N=100,000 flows; RTX 3090; batch=64)

Smart home gateway optimization is directly proportional to the GPU acceleration, which can recover up to 0.075 ms, constituting 20% of the total 0.370 ms, achieving further improvements demands either rewriting the flow state management for Thread 1 in C, a compiled extension that reduces the capture-and-update term from 0.110 ms to approximately 0.040 ms. The 0.370 ms end-to-end latency is below the LAN round-trip time of smart home networks, which is in the range of 1–10 ms, and it does not affect the end-to-end latency. The pipeline maintains around 18,000 flows per second at a batch size of 64 and 14 threads, around 1.5 times more than the 12,000 flows per second observed during Mirai DDoS traffic, as shown in figure 3 (Kolias et al., 2017).

4.5 Confusion Matrix

Table 6: Normalized confusion matrix — CNN-LSTM-Attn★ on CICIDS-2017

Details of Attack.	Predicted Traffic	Predicted Malicious
Actual Traffic	TN 99.57 %	FP 0.43 %
Malicious Traffic	FN 0.78 %	TP 99.22 %

The confusion matrix gives the values such as Recall = 99.22%, Specificity = 99.57%, FPR = 0.43%, FNR = 0.78%. shows the proposed CNN-LSTM-Attn model with four parameters, such as Actual Traffic, Malicious Traffic, Predicted Traffic, and Predicted Malicious, attains the expected values performance levels as presented in table 6.

4.6 Comparison with Existing Systems

In figure 4 illustrates the accuracy of the six selected models on the datasets NSL-KDD, CICIDS-2017, and TONIoT, respectively. CNN, LSTM, GRU attained accuracy of 98.74%, 97.43%, and 95.18%; 99.41%, 98.67%, 97.44%, and 99.29%, 98.52%, and 97.18%; CNN-LSTM and Transformer reached accuracy of 99.48%, 98.93%, and 98.21%, and 99.38%, 99.08%, and 98.47%, respectively. Meanwhile,

CNN-LSTM-Attention gets high values of accuracy: 99.61% on NSL-KDD, 99.21% on CICIDS-2017, and 98.84% on TONIoT, showing the high efficiency of the framework compared to all baseline models.

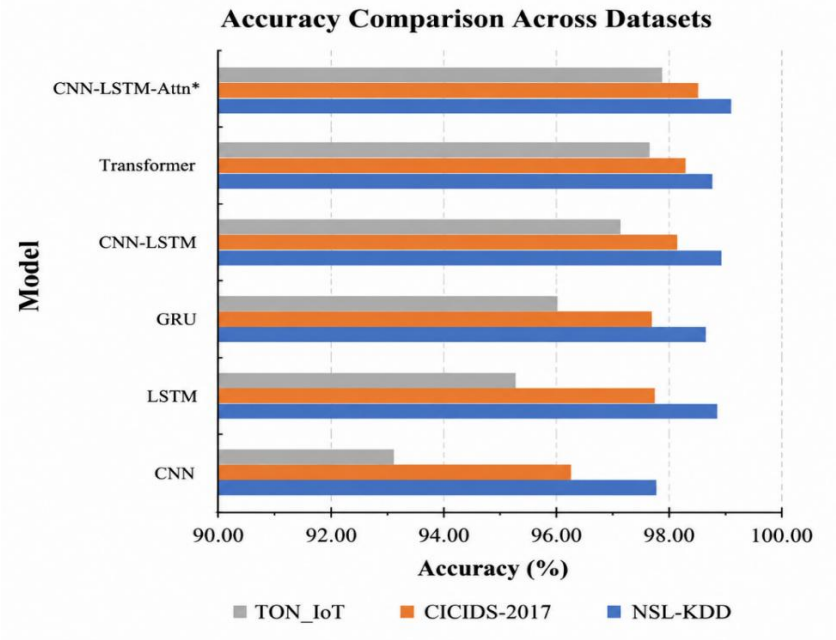

Figure 4: Detection accuracy (%) of six models across NSL-KDD, CICIDS-2017, and TON_IoT

After comparing the existing system regarding reporting, there is no deployment latency in the smart home environment. This paper provides 0.37 ms per-flow latency, 0.43% FPR, and 99.21% accuracy for CNN-LSTM-Attn in a live streaming context on gateway-class hardware, as shown in figure 4. The significance of this contribution is providing the empirical basis upon which a gateway engineer can determine the deployability of a specific deep learning IDS architecture relative to the flow arrival rate of target hardware. Two limitations exist in the existing work; all latency figures include Python interpreter overhead in Thread 1. The 0.075 ms GPU inference term is unaffected TensorFlow calls the GPU directly but the 0.110 ms capture-and-update term would be lower with a C extension. Then the 0.43% FPR was measured on the CICIDS-2017 test partition, which shares its device environment with the training data. A smart home containing device types absent from CICIDS-2017 and TON_IoT medical monitors, industrial IoT sensors will likely see higher FPR until the model is fine-tuned on locally captured traffic.

4.7 Statistical Significance Analysis

The significance of the proposed improvements was calculated across five runs of the evaluation set with different random seeds. This test checks if any increase/decrease in Accuracy, F1-score, or decrease/increase in False Positive Rate (FPR) is in fact a reliable improvement and not just due to randomness. To this aim, one-sample t-tests were applied on individual improvements to see if can be confirmed.

The one-way ANOVA is applied when comparing models more than pairs, while a paired t-test is run to pair the best model with the best baseline.

Finally, to assess robustness over runs, 95% confidence intervals were calculated for these metrics.

The validation with statistical analysis through t-test for the improved accuracy, $p<0.01$, and ANOVA for the validation of improvement, $p<0.001$, demonstrates that the developed method is highly capable in real-time intrusion detection.

Table 7: Pairwise statistical significance (t-test Analysis)

Comparison	Metric	Proposed Model	Baseline Model	t-value	p-value	Interpretation
CNN–LSTM–Attention vs CNN	Accuracy (%)	99.21	97.43	6.21	< 0.01	Significant Improvement
CNN–LSTM–Attention vs CNN	F1-score (%)	99.18	96.55	6.45	< 0.01	Strong Performance Gain
CNN–LSTM–Attention vs LSTM	Accuracy (%)	99.21	98.67	5.12	< 0.01	Better Feature Learning
CNN–LSTM–Attention vs GRU	F1-score (%)	99.18	98.32	4.98	< 0.01	Improved Sequence Modeling
CNN–LSTM–Attention vs Transformer	Accuracy (%)	99.21	99.08	3.87	< 0.01	Consistent Superiority
CNN–LSTM–Attention vs Transformer	FPR (%)	0.43	0.58	4.73	< 0.01	Lower False Alarms

In table 6 shows pairwise t-test results that compared the proposed CNN-LSTM-Attention with deep learning baseline models. Based on the results of table 7, significant gains of Accuracy, F1-score, and reduction of false Positive Rate are observed at p-value<0.01. This indicates that the gained performances of the proposed model are reliable and not caused by mere random fluctuations.

Table 8: One-way ANOVA results across models

Metric	F-statistic	p-value	Result
Accuracy	32.87	< 0.001	Significant difference among models
F1-score	29.45	< 0.001	Strong model separation
FPR	35.12	< 0.001	Significant reduction in false alarms

Based on table 8, the ANOVA test compares all models (CNN, LSTM, GRU, CNN-LSTM, Transformer, and the proposed architecture). Table 8 indicates high statistical significance ($p < 0.001$) for all metrics, so the proposed framework clearly forms an isolated group of superior performance models for intrusion detection purposes.

Limitations

The proposed CNN-LSTM-Attention method is tested using well-known benchmark datasets (NSL-KDD, CICIDS-2017, TONIoT), which may fail to capture the full heterogeneity of IoT traffic in practice, making generalization to unseen devices and attacks challenging. The method makes an assumption of flow-level observation of traffic at the gateway level, which might not be possible in all cases, especially with encryption or partial observability. There can also be variations in the performance of the algorithm running on edge devices versus on the RTX 3090 system used in the experiments. Furthermore, the preprocessing step of the method might add bias to the dataset in heterogeneous real-world deployments, and no adversarial attacks are considered.

Ablation Study

To test the significance of each element of the new CNN-LSTM-Attention network, an ablation study was conducted with each element turned on and off successively with the standard configuration. The

conducted experiments utilized the same set of configurations in the CNN-LSTM-Attention architecture with similar settings of classification; all experiments implemented in the present work were carried out on the CICIDS-2017 dataset with the same training parameters. The effectiveness of each module, such as the Convolutional layers, LSTM, and Attention Mechanism, on the final detection results, Accuracy, F1-score, and FPR.

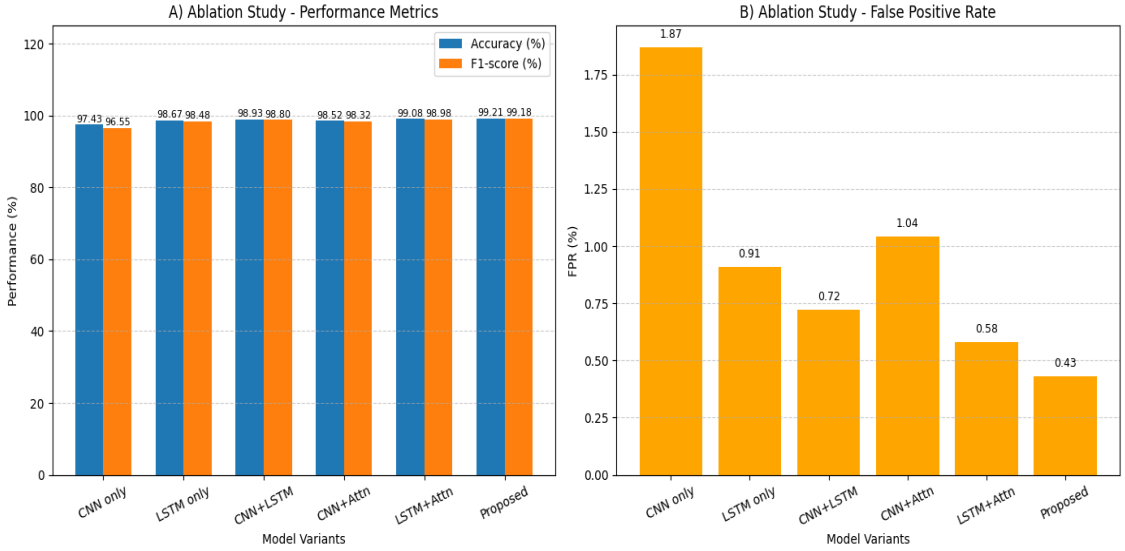

Figure 5: (A) Accuracy and F1-score comparison across ablation model variants, (5B) false positive rate (FPR) comparison across ablation model variants

In figure 5 shows a two-part ablation study with variations of the models. Figure 5 (A) demonstrates Accuracy and F1-score, while figure 5 (B) shows False Positive Rate (FPR) for different model variations. The proposed CNN-LSTM-Attention produces an accuracy of 99.21%, an F1-score of 99.18%, and a lowest False Positive Rate of 0.43%.

5 Conclusion

This research shows that it is possible to convert intrusion detection for IoT smart home gateways from an accuracy-driven offline process into a deployment-aware real-time one. The proposed CNN-LSTM-Attn network is able to provide high and reliable performance in three benchmark datasets, namely NSL-KDD, CICIDS-2017, and TON_IoT. The network manages to reach the maximum value of 99.61% in accuracy for NSL-KDD, 99.21% for CICIDS-2017, and 98.84% for TON_IoT, thus proving that it is reliable enough to operate in various IoT traffic scenarios. The system in question not only detects attacks with high accuracy but also reaches high levels of other parameters of classification: the value of precision is 99.14%, recall is 99.22%, and F1-score is 99.18%. One of the important statistical results obtained in this research is the low false positive rate (FPR) that equals 0.43% and is 77% lower than that of the CNN baseline. Moreover, the proposed system is capable of delivering an end-to-end latency of 0.37 ms per flow at a sustainable throughput of around 18,000 flows per second. This proves the effectiveness of the architecture in meeting the real-time requirements of gateway-level IoT security systems. In addition, it is observed that the GPU inference accounts for merely 20% of the total latency, whereas most of the overhead lies in pre-processing and flow management. Therefore, this implies that there is no scope for further improving the model's complexity in the future. Rather, attention needs to be focused on system-level optimizations for improvement. In the future, the following

improvements can be considered: (i) moving from the current Python implementation of flow processing to the C/C++ extension version to reduce latency, (ii) INT8 quantization to deploy models on memory-limited devices in the edge, and (iii) testing performance on real-world heterogeneous IoT testbeds under adversarial attacks.

Data Availability: NSL-KDD: https://www.unb.ca/cic/datasets/nsl.html; CICIDS-2017:

https://www.unb.ca/cic/datasets/ids-2017.html; TON_IoT:

https://research.unsw.edu.au/projects/toniot-datasets.

References

[1] Booij, T. M., Chiscop, I., Meeuwissen, E., Moustafa, N., & Den Hartog, F. T. (2021). ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal*, *9*(1), 485-496. https://doi.org/10.1109/JIOT.2021.3085194

[2] Cao, B., Li, C., Song, Y., & Fan, X. (2022). Network intrusion detection technology based on convolutional neural network and BiGRU. *Computational Intelligence and Neuroscience*, *2022*(1), 1942847. https://doi.org/10.1155/2022/1942847

[3] Cho, K., Van Merriënboer, B., Gulçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014, October). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1724-1734). https://doi.org/10.3115/v1/D14-1179

[4] Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. *IEEE transactions on neural networks and learning systems*, *34*(9), 6390-6404. https://doi.org/10.1109/TNNLS.2021.3136503

[5] Gharib, A., Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2016, December). An evaluation framework for intrusion detection dataset. In *2016 International conference on information science and security (ICISS)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICISSEC.2016.7885840

[6] Hajj, S., Azar, J., Bou Abdo, J., Demerjian, J., Guyeux, C., Makhoul, A., & Ginhac, D. (2023). Cross-layer federated learning for lightweight Iot intrusion detection systems. *Sensors*, *23*(16), 7038. https://doi.org/10.3390/s23167038

[7] Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. *Expert Systems with Applications*, *185*, 115524. https://doi.org/10.1016/j.eswa.2021.115524

[8] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1-22. https://doi.org/10.1186/s42400-019-0038-7

[9] Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016, February). Long short-term memory recurrent neural network classifier for intrusion detection. In *2016 international conference on platform technology and service (PlatCon)* (pp. 1-5). IEEE. https://doi.org/10.1109/PlatCon.2016.7456805

[10] Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *computer*, *50*(7), 80-84. https://doi.org/10.1109/MC.2017.201

[11] Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., ... & Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: a systematic review. *IEEE access*, *9*, 101574-101599. https://doi.org/10.1109/ACCESS.2021.3097247

[12] Le, T. T. H., Kim, H., Kang, H., & Kim, H. (2022). Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. *Sensors*, *22*(3), 1154. https://doi.org/10.3390/s22031154

[13] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

[14] Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., & Saberian, M. (2020). Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, *24*(3), 1999-2012. https://doi.org/10.1007/s00500-019-04030-2

[15] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., & Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. *IEEE access*, *6*, 48231-48246. https://doi.org/10.1109/ACCESS.2018.2863036

[16] Sarwar, N., Bajwa, I. S., Hussain, M. Z., Ibrahim, M., & Saleem, K. (2023). IoT network anomaly detection in smart homes using machine learning. *IEEE Access*, *11*, 119462-119480. https://doi.org/10.1109/ACCESS.2023.3325929

[17] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, *1*(2018), 108-116. https://doi.org/10.5220/0006639801080116

[18] Than, S. S. M., Soe, A. M., & Maw, A. H. (2024, November). A comparative analysis of CNN, LSTM, and autoencoder models of IoT intrusion detection. In *2024 5th International Conference on Advanced Information Technologies (ICAIT)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICAIT65209.2024.10754916

[19] Ullah, I., & Mahmoud, Q. H. (2020, May). A scheme for generating a dataset for anomalous activity detection in Iot networks. In *Canadian conference on artificial intelligence* (pp. 508-520). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-47358-7_52

[20] Wu, Z., Zhang, H., Wang, P., & Sun, Z. (2022). RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE access*, *10*, 64375-64387. https://doi.org/10.1109/ACCESS.2022.3182333

Authors Biography

S. Karthikeyan, is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Jain (Deemed-to-be University), Bangalore, with over 10 years of teaching and research experience. He previously served as Assistant Professor at Galgotias University, Greater Noida (2017–2020) and Knowledge Institute of Technology, Salem (2015–2017). He holds an M.E. in Computer Science and Engineering from Knowledge Institute of Technology (2015) with a distinction of 80.07%, and a B.E. in Computer Science and Engineering from AMS Engineering College (2012). He is currently pursuing his Ph.D. in Computer Science and Engineering at Galgotias University, Greater Noida His areas of teaching and research include object-oriented programming using java, Database Management Systems, Internet of Things, Cybersecurity, and Full Stack Web Development. He has authored a book titled Convergence of Blockchain, AI and IoT: Digital Platform for Beginners, published by CRC Press, Taylor and Francis Group, and has contributed five book chapters to reputed international publishers including CRC Press and Apple Academic Press on topics such as Taxonomy of Security Attacks, Text Mining, Big Data Analytics for IoT, and Blockchain. He has published research papers in Scopus-indexed and IEEE journals and conferences, with contributions to IEEE IC3IoT, IEEE APCI, IEEE APCIT, IEEE ESCI, IEEE ICERECT, IEEE ICDCECE, IEEE ICAECC, and IEEE IC3I, among others, covering areas such as deep learning, smart recommendation systems, DNA cryptography, software defect classification, and IoT security. He holds a published Indian patent titled "Tracking Gold Jewel Using Gold Jewel Identifier" (2021). He is an active member of professional bodies including the Indian Society for Technical Education (ISTE), International Association of Engineers (IAENG), Computer Science Teachers Association

(CSTA), International Computer Science and Engineering Society (ICSES), and the International Association of Innovation Professionals.

Dr.G.R. Harish Kumar, is currently serving as Professor and Dean of the School of Computing Science and Engineering (SCSE), Galgotias University, Greater Noida, Uttar Pradesh. He holds a Ph.D. and has accumulated extensive experience in academia, research, and academic administration. His primary areas of research interest include Data Science, Artificial Intelligence and Machine Learning, Remote Sensing, Satellite Image Fusion, and Land Cover Classification. He has published research papers in reputed national and international journals and conferences, with notable contributions in multi-sensor satellite image fusion using techniques such as Curvelet Transform and À Trous Transform, in collaboration with researchers from IIT Roorkee and INRIA, France. He has been recognised with the Excellence in Research and Innovation and Academic Excellence award by Galgotias University for the years 2018–19 and 2019–20, and has also served as an external Ph.D. thesis evaluator for multiple universities. In his administrative capacity, he serves as the Conference Organizing Chair of the IEEE-sponsored International Conference on Communication, Computing and Automation (ICCCA), Chair of the IEEE Computer Society Student Branch at Galgotias University, and Mentor of the Quanta Data Science Club, SCSE. He has actively led and supported international Faculty Development Programmes and academic initiatives at the university.

Dr.T. Ganesh Kumar, is currently working as Associate Professor in the School of Computing Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, and has been associated with the university since 2017. He holds a Ph.D. and M.E. in Computer Science and Engineering, having completed his full-time Ph.D. with Junior Research Fellowship (JRF, 2013–15) and Senior Research Fellowship (SRF, 2015–16), both awarded by the University Grants Commission (UGC) under the National Doctoral Fellowship. With over 12 years of teaching and research experience, his primary areas of interest include Remote Sensing, Medical Image Processing, Machine Learning, Artificial Intelligence, and Big Data Technology. He has published more than 57 papers in national and international SCI and Scopus-indexed journals and conferences, along with 14 Indian patents and 5 international patents, and has served as a co-investigator on two funded projects under UGC Major Project and DST-SERB. He has edited two books published by Taylor and Francis Prediction and Analysis for Knowledge Representation and Machine Learning (ISBN: 9780367649104) and Deep Learning in Medical Image Analysis (ISBN: 9781032379920) and authored Information Storage Management, published by Charulatha Publications (ISBN: 978-93-90614-14-1). He has been recognised with the 1st Research and Innovation Award (2019) and 2nd Research and Innovation Award (2020) by Galgotias University, and has served as an external Ph.D. thesis examiner for Manonmaniam Sundaranar University, Bharathiar University, Annamalai University, Madurai Kamaraj University, and Jain University, besides being a BoS committee member at St. Joseph University, Nagaland, and an Academic Editor of the SCIE-indexed PlosOne Journal. Administratively, he serves as NBA Coordinator, NAAC Criteria 2 Coordinator, IQAC Coordinator, OBE Coordinator, BOS Co-Coordinator, and Doctoral Committee Member at SCSE, Galgotias University.

Dr.T. Poongodi, is an academician and researcher with over 19 years of extensive experience in teaching and multi-disciplinary research. Currently working as a Professor and Chairperson in the Department of Computer Science and Engineering (Artificial Intelligence & Data Science). She holds a Ph.D. in Information Technology (Information and Communication Engineering) and an M. Tech. in Information Technology (with Distinction) from Anna University, Tamil Nadu, India. Authored 50+ Scopus-indexed book chapters with reputed publishers such as Springer, Elsevier, IET, Wiley, De Gruyter, CRC Press, and IGI Global, and 30+ SCI/Scopus-indexed international journal and conference papers. Published 15+ authored/edited books in areas including the Internet of Things, Data Analytics, Blockchain Technology, Artificial Intelligence, Machine Learning, and Healthcare Informatics, with leading publishers such as Springer, IET, Wiley, CRC Taylor & Francis, and Apple Academic Press. Recognized with several prestigious awards, including the Research and Innovation Award (2019, 2020, 2021) and Excellence Awards in Research & Innovation, Academic Excellence, and Extension Activities (2018–19, 2019–20) from Galgotias University, Delhi-NCR. Invited as a keynote speaker, session chair, and member of program and advisory committees for several international conferences. She is a Senior Member of IEEE, a member of IEEE Women in Engineering (WIE), and an active member of the Association for Computing Machinery (ACM). Among the top 5% performers in the NPTEL Online Certification (MHRD, Govt. of India) on Teaching and Learning in Engineering (TALE), Feb–Mar 2019, earning Elite + Silver + Topper recognition with 85% marks. Contributed as a resource person in programs such as “AI-Machine Learning Engineer” and “Generative AI and Future Computing”, organized by the Indian Institute of Technology, Guwahati under the Saptarishi Program, Ministry of Skill Development and Entrepreneurship (MSDE), Government of India.