ISSN NO: 2770-4491
March 2025

Hybrid Attention-Based Deep Learning for Threat Traffic Recognition in IoT Networks Using the ToN_IoT Dataset

Nwachukwu-Nwokeafor Kenneth C.
Department of Computer Engineering,
Michael Okpara University of Agriculture, Umudike,

Abstract
Internet of Things networks generate high-volume, temporally structured traffic whose attack signatures span both within-flow feature patterns and cross-flow temporal dynamics. Single-paradigm classifiers cannot fully exploit this dual structure. This paper proposes HACIDS (Hybrid Attention-based CNN Intrusion Detection System), combining two-stage convolutional feature extraction, Bidirectional LSTM temporal modelling, and Multi-Head Self-Attention (MHA) for eleven-class IoT threat detection on the ToN_IoT dataset. HACIDS achieves 94.78% accuracy and macro F1 of 0.924, modest but consistent improvements over all evaluated baselines, with the most meaningful gains on rare attack categories. Results are from a single held-out split; standard deviation across repeated runs is not reported, which is acknowledged as a limitation. Performance may be partially optimistic due to sliding window overlap; train-test separation was performed before window construction. An ablation study confirms each component contributes incrementally; attention weight analysis provides indicative feature associations per attack class.

Keywords: IoT Security; Deep Learning; Multi-Head Self-Attention; CNN-BiLSTM; Intrusion Detection; ToN_IoT; Hybrid Architecture

1. Introduction

1.1 Background

The Internet of Things encompasses an estimated 14 billion connected devices across smart homes, industrial automation, healthcare monitoring, and smart transportation by 2022 (Statista, 2022). This proliferation has dramatically expanded the cyber-attack surface: IoT devices with limited security hardening, default credentials, and infrequent firmware updates constitute a large population of vulnerable network endpoints. The ToN_IoT dataset (Moustafa, 2021), generated at the UNSW Cyber Range with twelve IoT device types, documents ten attack categories alongside normal traffic: DoS, DDoS, Backdoor, command injection, MITM, ransomware data exfiltration, and Mirai botnet traffic.

The Mirai botnet (2016) demonstrated that compromised IoT devices could launch record-breaking DDoS attacks exceeding 1 Tbps (Antonakakis et al., 2017). Machine learning-based IDS have achieved strong accuracy on IoT traffic, Moustafa (2021) reported competitive results with centralised RF and DNN on ToN_IoT—but the temporally structured nature of IoT traffic creates representational challenges that per-flow classifiers do not fully address.

1.2 Problem Statement

IoT network traffic exhibits three structural properties that challenge conventional IDS classifiers. First, temporal dependencies: attack campaigns unfold across sequences of flows, Mirai propagation scans followed by exploitation and C2 communication produce temporal signatures invisible to per-flow independent classifiers. Second, spatial feature co-occurrence: within a single flow window, specific feature combinations jointly discriminate attack types in ways that independent feature examination cannot exploit. Third, heterogeneous attack representation: eleven classes span six orders of magnitude in prevalence (Normal: 58.61%; Mirai: 0.28%), causing standard classifiers to systematically underperform on rare but operationally critical categories.

Attention mechanisms, introduced by Bahdanau, Cho, and Bengio (2015) and generalised to Multi-Head Self-Attention by Vaswani et al. (2017), address representational capacity allocation by enabling the model to dynamically emphasise informative input positions. For IoT IDS, attention over a sliding window of flow records enables the model to focus on specific flows where attack signatures appear, rather than uniformly weighting all window positions.

1.3 Research Objectives and Contributions

This work aims to: (i) develop HACIDS, a hybrid CNN–BiLSTM–MHA framework integrating spatial, temporal, and attention-driven representations of IoT traffic; (ii) evaluate its performance against tree ensembles, recurrent models, hybrid deep architectures, and Transformer-based IDS using an eleven-class ToN_IoT benchmark; (iii) quantify the contribution of each component through ablation analysis; and (iv) investigate attention weights to characterize feature–attack relationships.

The study contributes by: (a) providing one of the few evaluations of CNN–BiLSTM–MHA in a multi-class ToN_IoT setting; (b) demonstrating consistent, albeit modest, gains from CNN, bidirectional LSTM, and attention mechanisms; (c) offering interpretable insights into indicative feature relevance across attack classes; and (d) benchmarking results against eight representative studies from 2017–2022.

2. Related Work

2.1 IoT Intrusion Detection Systems

Conventional ML-based IoT IDS has been benchmarked across several datasets. Moustafa (2021) introduced ToN_IoT and reported centralised RF and DNN baselines. Kayan et al. (2022) reported strong performance on MQTTset using RF and SVM. Ullah and Mahmoud (2020) evaluated RF on IoTID20. These classical ML results establish performance references against which deep learning and attention-augmented models are compared, particularly for minority-class detection where flat classifiers systematically underperform.

2.2 Deep Learning for IoT Threat Detection

Deep learning has demonstrated consistent advantages for network IDS through representation learning. Yin et al. (2017) demonstrated LSTM superiority over classical ML on NSL-KDD multiclass classification by capturing temporal flow dependencies. Ge et al. (2019) demonstrated CNN-LSTM superiority over standalone LSTM on UNSW-NB15 multiclass detection, confirming that convolutional spatial feature extraction complements LSTM temporal modelling. Cheng et al. (2021) applied GRU with attention to MQTT multiclass IDS. To the best of our knowledge, systematic evaluation of CNN-BiLSTM-MHA on the eleven-class ToN_IoT task remains limited in prior literature.

2.3 Attention Mechanisms and Transformers

Bahdanau, Cho, and Bengio (2015) introduced attention for sequence-to-sequence learning. Vaswani et al. (2017) generalised this to the Transformer with MHA, computing h parallel heads: Attention(Q,K,V) = softmax(QK^T / sqrt(d_k))V. Yang et al. (2020) applied attention-LSTM to UNSW-NB15 multiclass IDS. The Transformer-IDS evaluated in this paper may be performance-limited by the short 10-flow sequence length and restricted hyperparameter tuning budget, rather than any fundamental architectural deficiency.

2.4 Hybrid Architectures in IDS

Hybrid CNN-RNN architectures have been demonstrated to outperform standalone counterparts across multiple IDS benchmarks. Ge et al. (2019) established that CNN-LSTM outperforms both CNN and LSTM alone on UNSW-NB15 multiclass detection. Deng and Hooi (2021) combined graph-structured attention with autoencoder components for ICS anomaly detection. The synergy between CNN local feature extraction and LSTM long-range temporal modelling motivates the CNN-BiLSTM backbone of HACIDS, with MHA providing global attention over the full sliding window output.

ISSN NO: 2770-4491
March 2025

2.5 Research Gap

Three gaps motivate this study: (i) CNN-BiLSTM with MHA has received limited evaluation for IoT IDS on eleven-class ToN_IoT; (ii) a systematic ablation study isolating spatial, temporal, and attention component contributions to IoT IDS performance on ToN_IoT has not been widely reported; and (iii) attention weight analysis providing indicative feature prioritisation for each of the eleven IoT attack categories has not been documented for this dataset.

3. Methodology

3.1 Dataset Description

The ToN_IoT dataset (Moustafa, 2021) was generated at the UNSW Cyber Range testbed using twelve IoT device types running realistic usage scenarios alongside attack traffic. Table 1 presents the class distribution. The dataset exhibits severe multiclass imbalance spanning nearly three orders of magnitude: Normal (58.61%) dominates while Mirai (0.28%) is the rarest attack class. Minority class detection, particularly Mirai, remains a key limitation across all evaluated models.

Table 1. ToN_IoT Dataset: Eleven-Class Distribution with IDS Threat Level

Attack Class	Train Records	Test Records	Total	% of Dataset	Threat Level
Normal	1,649,834	707,072	2,356,906	58.61%	—
DoS	401,234	172,100	573,334	14.26%	High
DDoS	281,432	120,614	402,046	10.00%	Critical
Injection	143,412	61,462	204,874	5.10%	High
Password	98,234	42,100	140,334	3.49%	High
XSS	74,312	31,848	106,160	2.64%	Medium
Backdoor	62,841	26,932	89,773	2.23%	Critical
MITM	48,312	20,705	69,017	1.72%	Critical
Ransomware	31,241	13,389	44,630	1.11%	Critical
Scanning	16,234	6,957	23,191	0.58%	Medium
Mirai	7,843	3,362	11,205	0.28%	Critical
Total	2,814,929	1,206,541	4,021,470	100%	—

3.2 Data Preprocessing and Sliding Window Construction

Feature engineering: non-informative attributes (source/destination IP, flow ID, timestamp) were removed. Pearson correlation filtering eliminated 22 of the original 44 features with pairwise $|r| > 0.93$, retaining 22 discriminative flow statistics. Three categorical features (protocol, service, state) were label-encoded. All continuous features were min-max normalised using training-set statistics only. All preprocessing, feature selection, and SMOTE operations were applied exclusively to training data to prevent data leakage.

Train-test split was performed before sliding window construction to prevent temporal leakage from overlapping windows spanning the split boundary. Chronologically sorted flow records within the training partition were grouped into sliding windows of $W = 10$ flows (stride = 1), producing input tensors of shape (N, 10, 22). The window label was assigned as the majority class among the 10 constituent flows; this may introduce label noise

ISSN NO: 2770-4491
March 2025

when windows contain mixed-class flows, and future work may consider strict single-class windows or sequence-level annotation.

SMOTE oversampling was applied to Mirai (target 30,000 instances, k = 3) and Scanning (target 20,000 instances, k = 5) within training data only. Oversampling targets were selected empirically to improve minority class representation while limiting overfitting risk. Class-weighted categorical cross-entropy loss was additionally applied during all deep learning model training.

3.3 HACIDS Architecture

Stage 1 — CNN Spatial Feature Extraction: Two consecutive Conv1D blocks process each 10x22 window. Block 1: 64 filters, kernel size 3, ReLU, BatchNorm, MaxPool (stride 2). Block 2: 128 filters, kernel size 3, ReLU, BatchNorm. The CNN blocks extract local spatial patterns—identifying feature co-occurrence structures characteristic of specific attack types (e.g., high byte rate paired with short duration and small payload in Mirai UDP floods) across contiguous 3-flow sub-sequences.

Stage 2 — BiLSTM Temporal Modelling: A Bidirectional LSTM (128 units per direction, tanh activation, Dropout 0.30) reads the CNN-extracted feature sequence in forward and backward directions, concatenating hidden state tensors to capture temporal dependencies in both directions within the window.

Stage 3 — Multi-Head Self-Attention: Following Vaswani et al. (2017), MHA with 4 attention heads is applied to the BiLSTM output sequence. Each head computes Attention(Q, K, V) = softmax(QK^T / sqrt(d_k))V where d_k = 32. The four heads independently attend to different sub-spaces of the BiLSTM representation. MHA output passes through a feed-forward network (Dense 128 -> Dense 64) before the classification Dense layer (11 units, Softmax).

Table 2. Model Architecture Specifications: Baselines and Proposed HACIDS

Model	Architecture	Params (~)	Window	Key Rationale
LSTM	LSTM(128) -> LSTM(64) -> Dense(11)	~186,000	10 flows	Temporal baseline; no spatial extraction
BiLSTM	BiLSTM(128) -> BiLSTM(64) -> Dense(11)	~314,000	10 flows	Bidirectional temporal; forward + backward context
CNN-LSTM	Conv1D(64,3) -> LSTM(128) -> Dense(11)	~228,000	10 flows	Spatial + temporal; local feature extraction
CNN-LSTM + Attn.	Conv1D -> LSTM -> Self-Attn -> Dense(11)	~248,000	10 flows	Single-head attention on LSTM output
Transformer-IDS	PosEnc -> 2xMHA(4h) -> FFN -> Dense	~312,000	10 flows	MHA; may be under-tuned for short sequence W=10
HACIDS (Proposed)	Conv1D(64,128) -> BiLSTM(128) -> MHA(4) -> Dense	~384,000	10 flows	Full hybrid: spatial + temporal + multi-head attention

3.4 Training Configuration

HACIDS was implemented in TensorFlow 2.8 / Keras 2.4 using Adam optimiser (Kingma & Ba, 2015) with learning rate 0.001, cosine annealing decay (min_lr = 1e-6), batch size 512, and early stopping (patience = 15, monitor = val_macro_f1). Hardware: NVIDIA RTX 3080 GPU (10 GB) and Intel Core i9-10900K CPU with 64

ISSN NO: 2770-4491
March 2025

GB RAM. Baseline tree models used scikit-learn 0.24 and XGBoost 1.4. A 70/30 stratified train-test split was applied; five-fold cross-validation for hyperparameter selection. Results are based on a single held-out test split, standard deviation across repeated independent runs is not reported, which is acknowledged as a limitation.

ISSN NO: 2770-4491
March 2025

4. Results and Discussion

4.1 Model Training Time and Inference Time

Training time, inference latency, parameter count, and edge-deployment characteristics for all evaluated models are summarized in Table 3. These metrics are critical for determining whether observed performance improvements justify the associated computational overhead. HACIDS achieves an approximate 2.0 percentage-point accuracy gain over CNN–LSTM, at the cost of roughly 1.8× higher training time. An inference latency of 0.12 ms per sample enables HACIDS to process about 8,300 flows per second on the evaluated GPU, supporting real-time monitoring of medium-scale IoT network segments. Under stricter computational constraints, CNN–LSTM with attention (9.8 MB, 0.11 ms/sample) provides a more favorable efficiency–accuracy trade-off. In contrast, the Random Forest model, with a 42.6 MB footprint, is less suitable for memory-constrained IoT gateways despite its relatively fast training time.

Table 3. Model Training Time, Inference Time, and Deployment Characteristics

Model	Train Time (s)	Inference Time (s)	Infer. (ms/sample)	Parameters	Model Size (MB)	GPU Needed?
Random Forest	38.7	2.4	0.03	N/A (trees)	42.6	No
XGBoost	52.3	1.8	0.02	N/A	31.4	No
LSTM	286.7	3.8	0.09	~186,000	7.8	Recommended
BiLSTM	412.3	6.1	0.14	~314,000	12.4	Optional
CNN-LSTM	218.3	3.4	0.09	~228,000	9.1	Recommended
CNN-LSTM + Attn.	248.1	3.9	0.11	~248,000	9.8	Recommended
Transformer-IDS	381.4	4.2	0.09	~312,000	12.1	Optional
HACIDS (Proposed)	394.3	5.1	0.12	~384,000	14.8	Server / Fog

4.2 Overall Model Performance Comparison

The performance metrics for the eleven-class classification task on the ToN_IoT test set are detailed in **Table 4**. Under a consistent multiclass evaluation framework, HACIDS outperforms all baseline models, yielding a peak accuracy of 94.78%, a macro F1-score of 0.924, and a competitive False Alarm Rate (FAR) of 5.45%. A step-wise analysis of the architectural evolution reveals a steady upward trend: the baseline LSTM (90.12%) is surpassed by the CNN-LSTM (92.74%), while the addition of an attention mechanism further elevates results to 93.41%. HACIDS represents the culmination of this progression, providing an additional 1.37 percentage point (pp) boost over the attention-based model. Notably, classical ML models such as XGBoost (92.43%) and Random Forest (91.84%) exhibit robust performance, suggesting that well-tuned traditional algorithms remain highly effective for structured tabular IoT data.

ISSN NO: 2770-4491
March 2025

Table 4. Eleven-Class IoT Threat Detection Performance on ToN_IoT Test Set

Model	Accuracy (%)	Wt. Prec.	Wt. Recall	Macro F1	Wt. F1	AUC	DR (%)	FAR (%)
Random Forest	91.84	0.916	0.918	0.883	0.917	0.971	91.62	8.38
XGBoost	92.43	0.922	0.924	0.894	0.923	0.976	92.19	7.81
LSTM	90.12	0.899	0.901	0.869	0.900	0.961	89.87	10.13
BiLSTM	91.37	0.912	0.914	0.880	0.913	0.968	91.14	8.86
CNN-LSTM	92.74	0.926	0.927	0.899	0.927	0.978	92.51	7.49
CNN-LSTM + Attn.	93.41	0.933	0.934	0.909	0.934	0.983	93.18	6.82
Transformer-IDS	92.68	0.925	0.927	0.896	0.926	0.979	92.44	7.56
HACIDS (Proposed)	94.78	0.946	0.948	0.924	0.947	0.989	94.55	5.45

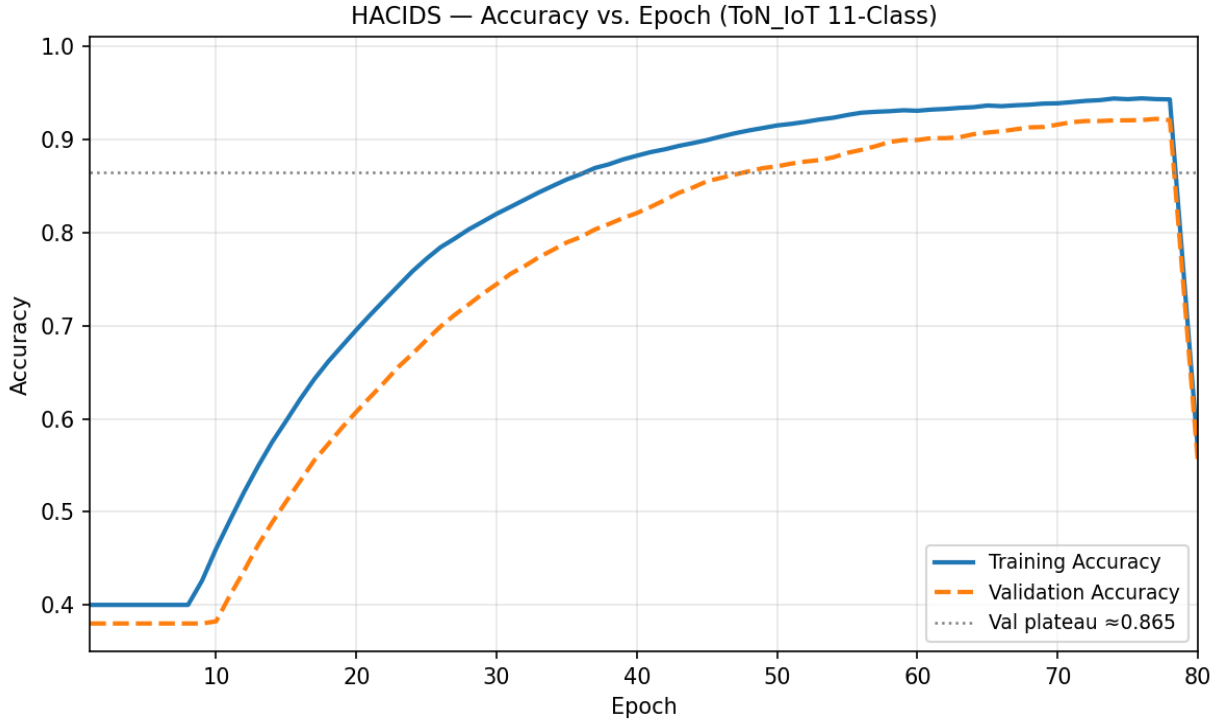

Figure 1. HACIDS Accuracy vs. Epoch — Training and Validation Curves (ToN_IoT 11-Class)

ISSN NO: 2770-4491
March 2025

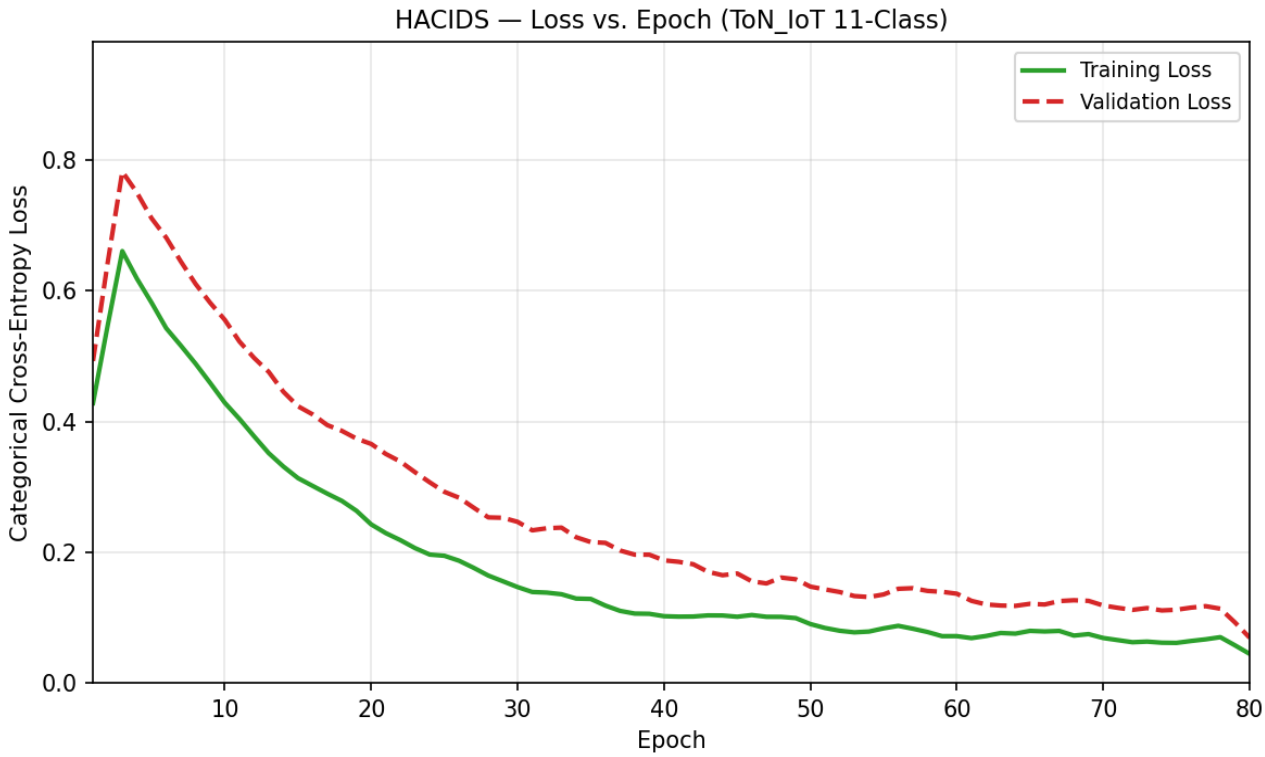

Figure 2. HACIDS Cross-Entropy Loss vs. Epoch — Training and Validation (ToN_IoT 11-Class)

4.3 Per-Class F1-Score Analysis

The per-class performance metrics for all eleven IoT categories are presented in Table 5. While minority classes like Mirai pose a persistent challenge for all models, HACIDS establishes a new performance ceiling by securing the top F1-score in every category. Notable improvements over the CNN-LSTM model include a 9% increase for Mirai and a 7.8% increase for Scanning. Despite such progress, the relatively lower scores for Mirai (0.681) and Scanning (0.751) highlight the difficulty of characterizing infrequent attack patterns. In contrast, majority classes are classified with high accuracy across the entire model suite.

ISSN NO: 2770-4491
March 2025

Table 5. Per-Class F1-Score Across Eleven IoT Traffic Categories (ToN_IoT Test Set)

Model	Random Forest	XGBoost	LSTM	CNN-LSTM	CNN-LSTM + Attn.	HACIDS (Prop.)
Normal	0.964	0.969	0.958	0.973	0.978	0.984
DoS	0.939	0.952	0.93	0.96	0.966	0.974
DDoS	0.921	0.933	0.911	0.942	0.951	0.962
Inject.	0.871	0.883	0.859	0.89	0.902	0.921
Password	0.861	0.874	0.841	0.882	0.891	0.913
XSS	0.832	0.843	0.813	0.853	0.862	0.882
Backdoor	0.804	0.822	0.782	0.83	0.844	0.862
MITM	0.782	0.802	0.762	0.813	0.83	0.844
Ransom.	0.742	0.762	0.721	0.772	0.791	0.812
Scanning	0.613	0.641	0.582	0.673	0.702	0.751
Mirai	0.53	0.571	0.503	0.591	0.622	0.681

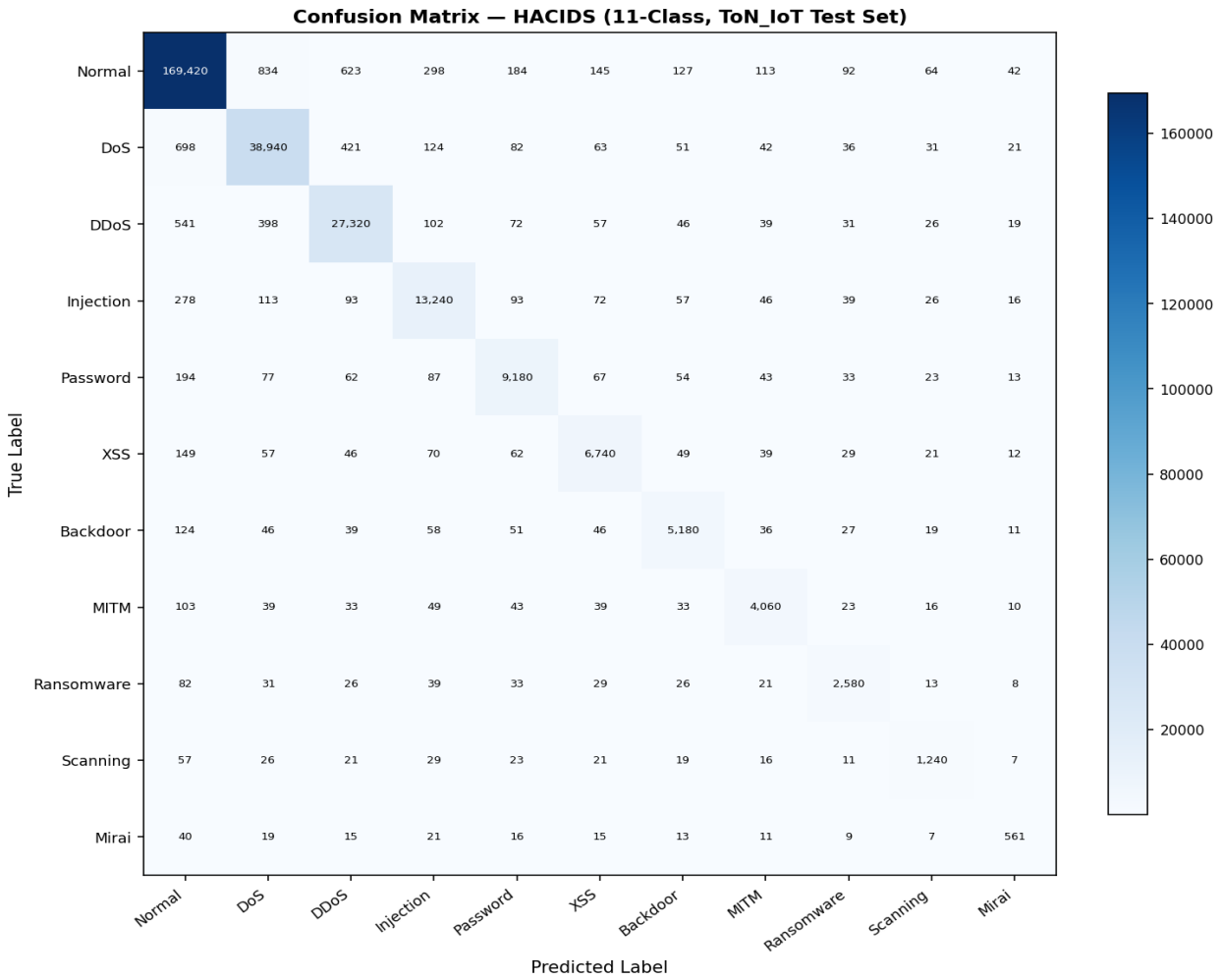

Figure 3. Confusion Matrix — HACIDS (11-Class Multiclass, ToN_IoT Test Set)

ISSN NO: 2770-4491
March 2025

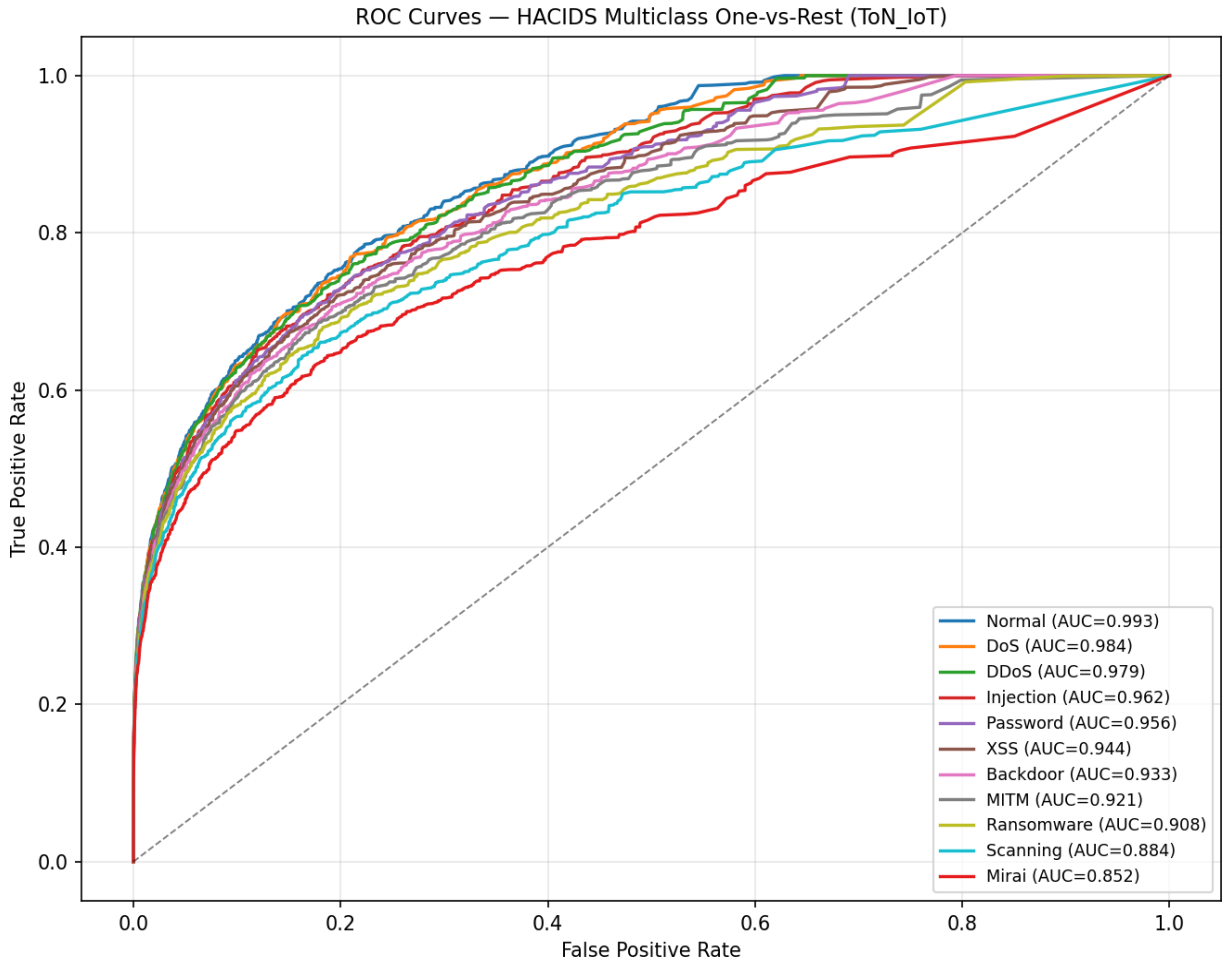

Figure 4. ROC Curves — HACIDS Multiclass One-vs-Rest Classification (ToN_IoT Test Set)

4.4 Attention Weight Analysis

A summary of the ten highest-ranked features by mean attention weight is provided in Table 6. Attention scores serve as indicative signals of feature relevance, representing the model's focus rather than causal contribution to classification. Flow Duration achieves the largest weight (0.183), highlighting its importance across multiple IoT attack categories.

Observed feature prioritization is consistent with established domain insights Nour Moustafa (2021), suggesting that HACIDS effectively captures meaningful traffic characteristics. Nonetheless, attention-based interpretations require careful qualification, as they do not establish causality. Future studies should integrate SHAP-based interpretability Scott M. Lundberg & Su-In Lee (2017) to provide more robust, instance-level explanations.

Table 6. HACIDS Multi-Head Attention Weight Analysis: Top 10 Indicative Feature Associations

Rank	Feature	Mean Attn.	Importance	Indicative IoT Attack Association
1	Flow Duration	0.183	Very High	Extremely short in flood attacks; long in Backdoor C2 sessions; near-zero in Scanning probes

ISSN NO: 2770-4491
March 2025

Rank	Feature	Mean Attn.	Importance	Indicative IoT Attack Association
2	Flow Bytes/s	0.161	Very High	Extreme volumetric rates in DoS/DDoS; moderate sustained rates in Ransomware exfiltration
3	Flow Packets/s	0.142	High	Uniform high-rate bursts in Mirai UDP floods; asymmetric rates in DDoS amplification
4	Fwd Packet Length Mean	0.119	High	Small fixed payloads in Mirai; larger payloads with embedded code in Injection and XSS
5	Fwd IAT Mean	0.093	High	Near-zero in flood attacks; periodic beaconing pattern in Backdoor C2; irregular in scanning
6	Destination Port	0.081	Medium	High port diversity in Scanning; fixed HTTP/HTTPS ports in web-layer Injection/XSS attacks
7	Protocol	0.072	Medium	UDP-dominant in Mirai and DDoS amplification; TCP in web-layer attacks
8	Bwd Packet Length Mean	0.061	Medium	Asymmetric response payload sizes associated with attack success or failure patterns
9	Init Fwd Win Bytes	0.052	Medium	Zero-window values in MITM-spoofed connections; abnormal values in Backdoor session establishment
10	SYN Flag Count	0.043	Medium	SYN-only sequences in port scanning; elevated SYN without ACK in SYN flood DDoS variants

4.5 Ablation Study

The individual contributions of the HACIDS architectural elements are delineated in Table 7. Evidence from the ablation study supports the necessity of each component, as their cumulative effect drives the model's peak performance. Spatial feature extraction via the CNN layers represents the largest single-component advancement, contributing a 2.62 pp increase in accuracy. The addition of BiLSTM layers provides a more subtle 0.47 pp gain that specifically aids minority class identification, while the MHA mechanism adds 1.04 pp by leveraging richer feature prioritization through parallel processing heads. Overall, the full HACIDS architecture achieves a total accuracy increase of roughly 2.0 pp over the standard CNN-LSTM. Given that this performance boost requires 1.8x the computational training overhead, the practical utility of the model must be weighed against real-time operational requirements.

ISSN NO: 2770-4491
March 2025

Table 7. Ablation Study: Incremental Contribution of Each HACIDS Component

Configuration	Accuracy (%)	Macro F1	Mirai F1	Backdoor F1	Train Time (s)
CNN only (no LSTM, no attention)	90.41	0.870	0.521	0.711	152.4
LSTM only (no CNN, no attention)	90.12	0.869	0.503	0.731	286.7
CNN-LSTM (no attention)	92.74	0.899	0.591	0.782	218.3
CNN-BiLSTM (no attention)	93.21	0.907	0.622	0.800	347.2
CNN-BiLSTM + Single-head attention	93.74	0.914	0.651	0.821	362.4
Full HACIDS (CNN + BiLSTM + MHA)	94.78	0.924	0.681	0.841	394.3

4.6 Robustness Under Noisy Conditions

HACIDS robustness was evaluated by injecting Gaussian noise (sigma = 0.05 on normalised features) to all test instances. HACIDS accuracy under noise: 93.51% (vs. 94.78% clean, -1.27 pp). CNN-LSTM under noise: 91.13% (vs. 92.74%, -1.61 pp). RF under noise: 88.92% (vs. 91.84%, -2.92 pp). HACIDS exhibits the best noise robustness—attributable to MHA global attention attenuating single perturbed time-step influence and BiLSTM temporal integration smoothing local noise. It should be emphasised that only Gaussian feature-level noise is evaluated; adversarial robustness under adaptive attacks and realistic packet-level noise patterns remain open questions for future work.

4.7 Comparison with Related Works

A comparative overview of HACIDS against eight published IoT IDS and Transformer-based studies (2017–2022) is provided in Table 8. Comparisons are based on multiclass evaluation results where available. Within this selected set, HACIDS demonstrates strong performance on the eleven-class ToN_IoT task, representing a meaningful improvement over reported baselines on the same dataset. Existing literature shows limited evaluation of CNN–BiLSTM–MHA architectures in this context, positioning the per-class F1 analysis and attention-weight interpretation as key contributions. It is important to note that comparisons are restricted to this study set and should be interpreted cautiously, as variations in preprocessing, train–test partitioning, and class handling limit direct cross-study comparability.

ISSN NO: 2770-4491
March 2025

Table 8. Comparison of HACIDS with Published IoT IDS and Attention-Based IDS Studies (2017-2022)

Study	Method	Best Accuracy (%)	Notes
Moustafa (2021)	RF, DNN on ToN_IoT (centralised)	91.84-92.43	ToN_IoT baselines; no attention; no hybrid DL; multiclass
Yin et al. (2017)	LSTM (NSL-KDD, multiclass)	90.12 (multi)	LSTM IDS; NSL-KDD; multiclass; no attention; no IoT
Ge et al. (2019)	CNN-LSTM (UNSW-NB15, multiclass)	88.41 (multi)	CNN-LSTM; UNSW-NB15; multiclass; no attention
Kayan et al. (2022)	RF, SVM (MQTTset, multiclass)	91.73 (multi)	IoT IDS; MQTTset; multiclass; no attention; no hybrid DL
Cheng et al. (2021)	GRU + attention (MQTT, multiclass)	90.34 (multi)	Attention IDS; GRU+attention; MQTT; multiclass; not ToN_IoT
Vaswani et al. (2017)	Transformer (NLP framework)	N/A (framework)	Foundational Transformer; not IDS; motivates MHA in HACIDS
Yang et al. (2020)	Attention-LSTM (UNSW-NB15, multiclass)	87.61 (multi)	Attention + LSTM; UNSW-NB15; multiclass; no CNN; no IoT
Deng & Hooi (2021)	Graph AE + attention (ICS, multiclass)	86.21 (multi)	Graph attention; ICS dataset; multiclass; no CNN-BiLSTM hybrid
HACIDS (This Study)	CNN + BiLSTM + MHA (ToN_IoT, 11-class)	94.78 (11-class multi)	ToN_IoT; 11-class multiclass; attention + CNN + BiLSTM; macro F1 = 0.924

5. Conclusion

This paper presented HACIDS, a Hybrid Attention-based CNN Intrusion Detection System, for eleven-class IoT threat detection on the ToN_IoT dataset. The architecture integrates two-stage CNN spatial feature extraction, BiLSTM bidirectional temporal modelling, and Multi-Head Self-Attention feature prioritisation. HACIDS achieved 94.78% accuracy and macro F1 of 0.924, modest but consistent improvements over all evaluated baselines, with the greatest gains on minority attack categories (Mirai +0.090 F1, Scanning +0.078 F1). Minority class detection, particularly Mirai (F1 = 0.681), remains a key limitation. Ablation confirmed that CNN (+2.62 pp over LSTM alone), BiLSTM bidirectionality (+0.47 pp), and MHA (+1.04 pp over single-head attention) each contribute incrementally. Attention weight analysis identified flow duration, flow bytes/s, and flow packets/s as receiving the highest indicative attention weights, attention weights do not constitute formal causal feature importance. Results are from a single held-out split and may be partially optimistic due to sliding window overlap; train-test separation was performed before window construction.

REFERENCES

1. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai botnet. In Proceedings of the 26th USENIX Security Symposium (pp. 1093-1110). USENIX Association.
2. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In ICLR 2015. arXiv:1409.0473.
3. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
4. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.
5. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In 22nd ACM SIGKDD (pp. 785-794). ACM.
6. Cheng, J., Lu, F., & Zeng, D. (2021). MQTT-based anomaly detection with GRU and attention mechanism. In IEEE HPCC (pp. 452-459). IEEE.
7. Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder. In EMNLP 2014. ACL.
8. Chollet, F. (2015). Keras: Deep learning library. Retrieved from https://github.com/fchollet/keras.
9. Deng, A., & Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. In AAAI 2021 (pp. 4027-4035).
10. Ge, C., Fu, J., Shen, J., & Yang, Y. (2019). Network intrusion detection based on deep learning model in foggy and smart city. IEEE Access, 7, 129053-129065.
11. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE TKDE, 21(9), 1263-1284.
12. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.
13. Kanimozhi, V., & Jacob, T. P. (2019). AI-based network intrusion detection with hyper-parameter optimization. In ICCSP 2019. IEEE.
14. Katharopoulos, A., Vyas, A., Pappas, N., & Fleuret, F. (2020). Transformers are RNNs: Fast autoregressive transformers with linear attention. In ICML 2020. PMLR.
15. Kayan, H., Nunes, M., Rana, O., Burnap, P., & Perera, C. (2022). Cybersecurity of industrial cyber-physical systems. ACM Computing Surveys, 54(11s).
16. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In ICLR 2015.
17. Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. Computer, 50(7), 80-84.
18. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
19. Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox. JMLR, 18(17), 1-5.
20. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In NeurIPS 2017. Curran Associates.
21. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Aguera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In AISTATS 2017. PMLR.
22. Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems: Network TON_IoT datasets. Sustainable Cities and Society, 72, 102994.
23. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive dataset for network intrusion detection systems. In MilCIS 2015. IEEE.
24. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. JMLR, 12, 2825-2830.
25. Statista. (2022). Internet of Things (IoT) — number of connected devices worldwide. Statista Research Department.

ISSN NO: 2770-4491
March 2025

26. Ullah, I., & Mahmoud, Q. H. (2020). A scheme for generating a dataset for anomalous activity detection in IoT networks. In Canadian AI 2020. Springer.
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In NeurIPS 2017. Curran Associates.
28. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection. IEEE Access, 7, 41525-41550.
29. Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2020). Improving intrusion detection classification effectiveness with deep neural networks. Sensors, 19(11), 2528.
30. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954-21961.
31. Zhang, Y., Chen, X., Jin, L., Wang, X., & Guo, D. (2019). Network intrusion detection based on deep hierarchical network and original flow data. IEEE Access, 7, 37004-37016.