future internet

MDPI

Article

Fine-Grained IoT Attack Classification Using a Cross-Attention CNN-BiLSTM Model

Mohamed Ali Fakri [1,*], Abdellah Najid [1], Rachid Ben Said [2] and Nezha El Idrissi [1]

[1] AGNOX Research Group, National Institute of Posts and Telecommunications, Rabat 10100, Morocco; najid@inpt.ac.ma (A.N.); nezha.elidrissi10@gmail.com (N.E.I.)
[2] Department of Computer Engineering, Ankara University, Ankara 06830, Turkey; benrachidsaid@gmail.com
* Correspondence: fakri.mohamedali@doctorant.inpt.ac.ma

Abstract

Deep learning intrusion detection systems perform well when traffic is merely separated into normal and malicious, but fine-grained recognition of the specific attack family remains difficult in Internet of Things (IoT) environments because of severe class imbalance and overlapping feature distributions. This work proposes an attention-enhanced CNN-BiLSTM fusion model for the multi-class classification of IoT attacks over eight families. A convolutional branch extracts local feature interactions, whereas a bidirectional Long Short-Term Memory (BiLSTM) branch reads the standardized flow descriptor as an ordered sequence and encodes dependencies among non-adjacent feature segments in both directions. Multi-head self-attention and a bidirectional cross-attention block let the two representations interact dynamically and suppress redundant information. Class imbalance is addressed with a focal loss and class-balanced weighting. The framework was evaluated on the large-scale CIC-IoT2023 benchmark under an eight-class taxonomy. On more than 3.5×10^5 test flows, the proposed model reached 99.06% accuracy and a 98.99% weighted F1-score, outperforming standalone CNN, LSTM, CNN-LSTM, and CNN-BiLSTM baselines retrained on the same corrected data pipeline under an identical objective and budget.

Keywords: CNN-BiLSTM fusion; cross-attention; multi-class intrusion detection; focal loss; class imbalance; CIC-IoT2023; IoT security; deep learning

1. Introduction

The wide adoption of Internet of Things (IoT) devices in industrial control systems, home automation deployments, and critical infrastructure has considerably increased the attack surface of contemporary networks [1]. Modern IoT infrastructures must therefore be prepared to face a variety of cyberattacks, including volumetric Distributed Denial of Service (DDoS), Mirai-like botnets, reconnaissance, credential cracking, and ransomware-driven exfiltration [2,3]. The exact identification of the attack family is a key step in any efficient response [4].

Contemporary Intrusion Detection Systems (IDSs) usually limit themselves to the binary classification of normal versus abnormal traffic [5–8]. Although this is appropriate for alarm generation, it is of limited operational utility, since forensic analysts and automated defense systems need fine-grained labels in order to apply targeted countermeasures [4,9]. Extending the task to multiple categories is not trivial. Modern IoT datasets are highly imbalanced: volumetric attacks dominate, whereas application-layer threats such as

Check for updates

Academic Editors: Eszter Kail, Anna Bánáti and Miklós Kozlovszky

Received: 17 August 2026
Revised: 22 September 2026
Accepted: 24 September 2026
Published: 29 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

cross-site scripting, SQL injection, or credential cracking are rare [2,10]. Classifiers trained with the usual categorical cross-entropy loss consequently favor the majority classes and under-report rare but operationally important attacks [9].

Deep learning has shown strong potential for the binary IDS task [11–16]. Convolutional Neural Networks (CNNs) excel at extracting spatial correlations from structured traffic representations [11,13], whereas recurrent architectures such as Long Short-Term Memory (LSTM) capture temporal dependencies in sequential traffic flows [14,17]. Hybrid CNN-LSTM architectures [18–23] jointly leverage the spatial and the temporal views, and attention-based extensions [24–28] have further increased detection accuracy. Most of these studies, however, were evaluated in the binary setting [9], leaving the more demanding fine-grained multi-class scenario only partially explored.

Several recent studies have nevertheless addressed the multi-class problem. Authors in [6] proposed a two-stage classifier ensemble evaluated on multi-class NSL-KDD and UNSW-NB15, showing that ensembles outperform single classifiers when the attack taxonomy is fine-grained. In [8], a Gini-impurity-based weighted random forest was applied to UNSW-NB15 and ToN-IoT for multi-class attack categorization, but the approach relies on hand-crafted feature selection. Ten supervised and unsupervised algorithms were benchmarked on the multi-class CICIDS2017 dataset in [7], confirming that classical machine learning suffers markedly from class imbalance. A bidirectional LSTM was used in [15] for multi-class detection on NSL-KDD, but the severe class skew typical of modern IoT datasets was not addressed [2,10].

Class imbalance is a well-recognized obstacle in multi-class IDS [9]. Common mitigation strategies include over-sampling, under-sampling, cost-sensitive learning, and the focal loss originally proposed for dense object detection. Class-balanced cross-entropy and weighted loss formulations have been explored for IDS [2,29], but the combination of a focal loss with class-balanced weighting inside a cross-attention CNN-BiLSTM architecture has not, to the best of the authors' knowledge, been investigated.

A small number of works have explored bidirectional recurrent units within hybrid IDS frameworks. It was demonstrated in [15] that BiLSTM consistently improves over unidirectional LSTM in multi-class settings. A CNN-BiLSTM with hybrid feature selection was introduced in [21] for software-defined networking environments, achieving strong binary results, whereas CNN-LSTM detection was validated across multiple datasets in [20] and an intelligent regularized CNN-LSTM model was proposed in [23]. Attention mechanisms [24–28,30] have also been increasingly applied to IDS in order to selectively emphasize informative features: a transformer-based IDS was proposed in [25], transformer architectures were applied to cloud-security intrusion detection in [27], and an attention-based CNN-LSTM outperforming non-attentive baselines was reported in [28]. None of these works, however, integrates cross-attention between a convolutional and a bidirectional feature-sequence branch in the multi-class setting addressed here.

To address the above gap, this paper proposes an imbalance-aware cross-attention CNN-BiLSTM framework for fine-grained multi-class IoT intrusion detection, built around three complementary design principles. First, a BiLSTM branch reads the standardized feature vector as an ordered sequence and encodes it simultaneously in the forward and the backward direction. CIC-IoT2023 flow descriptors have no inherent temporal order, and a recurrent network, even a bidirectional one, remains sensitive to the order of its inputs; the native column order of the descriptors is therefore an architectural assumption of the model. The bidirectional encoding is intended to relate non-adjacent feature segments that lie outside the local receptive fields of the convolutional branch and to provide context for the cross-attention layer. Second, instead of the conventional categorical cross-entropy, the model is trained with a focal loss combined with class-balanced weighting, which

explicitly mitigates the dominance of majority attack families and focuses the gradient on hard-to-classify minority samples. Third, the framework is evaluated under an eight-class attack-family taxonomy derived from the fine-grained labels of CIC-IoT2023 [10], which allows a uniform empirical comparison with strong deep learning baselines.

The main contributions of this work are the following:

- A cross-attention CNN-BiLSTM fusion architecture for fine-grained multi-class IoT intrusion detection that combines convolutional local feature extraction with bidirectional feature-sequence modeling and bidirectional cross-attention.
- An imbalance-aware training strategy that combines a focal loss with class-balanced weighting, designed for severely imbalanced multi-class IoT attack datasets.
- A unified eight-class attack taxonomy spanning Normal, DDoS, DoS, Reconnaissance, Spoofing/MITM, Web Attacks, Mirai Botnet, and Brute Force, derived from the fine-grained labels of CIC-IoT2023 [10].
- An extensive comparative evaluation against four deep learning baselines (CNN, LSTM, CNN-LSTM, and CNN-BiLSTM) in which the proposed model obtains the highest accuracy and macro F1-score among the compared deep learning models, although the experiments do not isolate which architectural component is responsible for this difference.

Figure 1 summarises the stages an intruder typically follows when penetrating an IoT network, together with the attack family that each stage produces in the observed traffic. Detecting only whether traffic is malicious conceals this progression, whereas recognising the specific family indicates which stage of the intrusion is in progress.

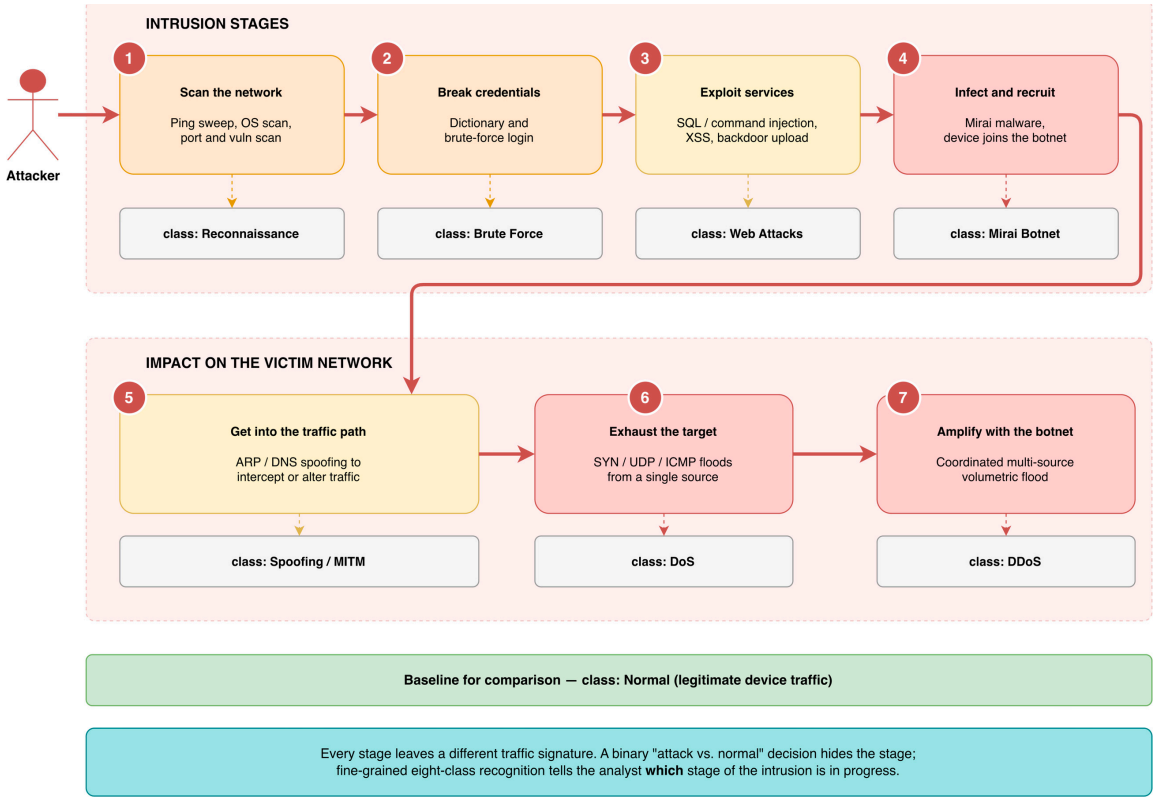


Figure 1. Stages of an intrusion into an IoT network and the attack family that each stage produces in the observed traffic.

2. Methodology

The complete detection workflow is shown in Figure 2. Traffic is mirrored at the edge gateway and converted into flow records, which are cleaned, standardised and reshaped before being classified by the proposed model; the predicted attack family and its confidence are then reported to the security operations centre.

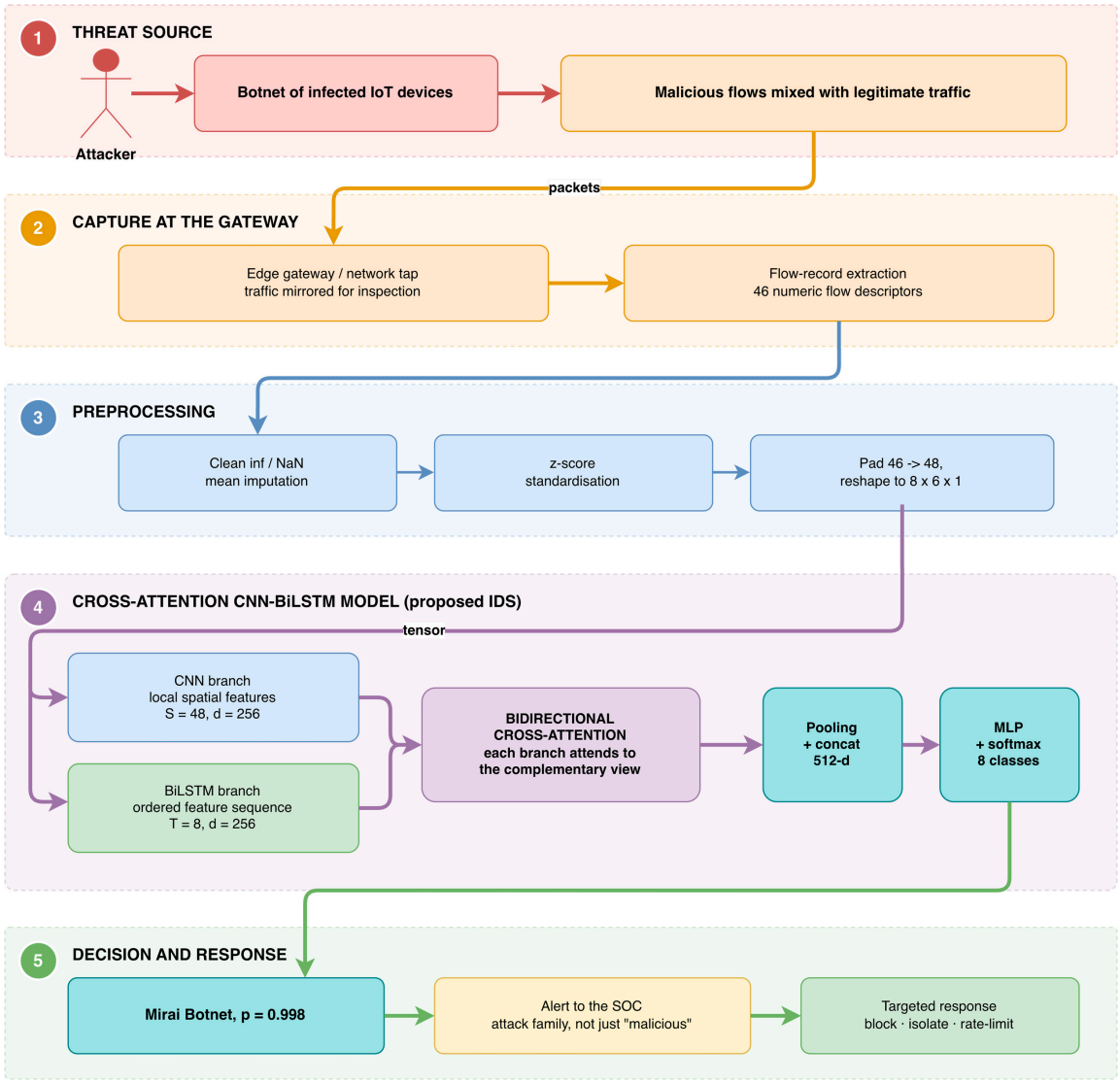


Figure 2. End-to-end detection workflow, from traffic capture at the edge gateway to the alert raised by the proposed cross-attention CNN-BiLSTM model.

2.1. Cross-Attention Mechanism

Attention mechanisms [30] compute a weighted aggregation of value vectors based on the similarity between a query and a set of keys, as in

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where d_k is the dimensionality of the keys. In multi-head attention, the operation is carried out in parallel over several projection subspaces and the outputs are concatenated and linearly projected [30].

In the proposed framework, bidirectional cross-attention enables a dynamic interaction between the local features extracted by the CNN branch and the sequential features

produced by the BiLSTM branch [28,31]. Let $\mathbf{F}_{\text{CNN}} \in \mathbb{R}^{S\times d}$ denote the CNN feature map reshaped into a sequence of S positions of dimension d, and $\mathbf{F}_{\text{BiLSTM}} \in \mathbb{R}^{T\times d}$ the BiLSTM output sequence of T feature steps, each of dimension d. Both representations share the same channel dimension d = 256, so that the CNN sequence (S = 48) and the BiLSTM sequence (T = 8) can be aligned through cross-attention without any additional projection.

In the first cross-attention direction, given by (2), the CNN representation serves as the query while the BiLSTM representation provides the keys and the values, allowing the convolutional branch to selectively attend to the most relevant sequential patterns.

$$A_{CNN\leftarrow BiLSTM} = \text{softmax}\left(\frac{Q_{CNN}{K_{BiLSTM}}^T}{\sqrt{d_k}}\right)V_{BiLSTM} \tag{2}$$

In the reverse direction, given by (3), the BiLSTM representation serves as the query while the CNN representation provides the keys and the values, allowing the sequential branch to focus on the most discriminative local features.

$$A_{BiLSTM\leftarrow CNN} = \text{softmax}\left(\frac{Q_{BiLSTM}{K_{CNN}}^T}{\sqrt{d_k}}\right)V_{CNN} \tag{3}$$

Residual connections followed by layer normalization are applied after each cross-attention operation [32], as in (4) and (5).

$$\widetilde{F}_{CNN} = \text{LayerNorm}(F_{CNN} + A_{CNN\leftarrow BiLSTM}) \tag{4}$$

$$\widetilde{F}_{BiLSTM} = \text{LayerNorm}(F_{BiLSTM} + A_{BiLSTM\leftarrow CNN}) \tag{5}$$

Each attended sequence is finally reduced to a fixed-length vector by global average pooling along the sequence axis, which yields two 256-dimensional descriptors.

2.2. Focal Loss for Imbalanced Multi-Class Classification

In multi-class IDS, severe class imbalance drives conventional cross-entropy training towards a classifier biased in favor of the majority classes [9]. To mitigate this effect, the proposed framework adopts the focal loss of (6), which down-weights well classified examples and concentrates the gradient on hard, and typically minority, samples.

$$\text{L}_{focal} = -\alpha(1-p_t)^{\gamma}\log(p_t) \tag{6}$$

where p_t is the predicted probability of the ground-truth class, the focusing parameter γ controls the rate at which easy examples are down-weighted, and the balancing factor α scales the whole term. Values of $\gamma = 2$ and $\alpha = 0.25$ were used in the experiments. In addition, class-balanced weights derived from the inverse frequency of each class in the training data are applied during training in order to further amplify the contribution of the minority classes, as in

$$w_c = \frac{N}{C\cdot n_c} \tag{7}$$

where N is the total number of training samples, C is the number of classes, and n_c is the number of samples of class c.

2.3. Dataset

The empirical evaluation was conducted on the CIC-IoT2023 benchmark [10], a large-scale intrusion detection dataset that covers heterogeneous IoT traffic scenarios and a broad spectrum of cyberattacks, ranging from volumetric DDoS to application-layer attacks [2].

The dataset is publicly released by the Canadian Institute for Cybersecurity under the terms stated by its owners, and it has already been used as an evaluation benchmark in [33,34]. It contains traffic collected from a controlled IoT testbed populated with tens of real devices and provides fine-grained labels for an extensive set of attacks, including multiple DDoS variants (ICMP, UDP, TCP, SYN, PSHACK, and RSTFIN), DoS, Reconnaissance (ping sweep, OS scan, port scan, and vulnerability scan), Mirai (greeth flood, greip flood, and UDP-plain), Spoofing/MITM (ARP and DNS), Web Attacks (SQL injection, command injection, cross-site scripting, and backdoor), and Brute Force. These fine-grained categories were mapped onto a unified eight-class taxonomy, namely Normal, DDoS, DoS, Reconnaissance, Spoofing/MITM, Web Attacks, Mirai Botnet, and Brute Force, by merging the variants of each high-level family. The resulting distribution is shown in Figure 3 and is extremely skewed, which is precisely the setting the proposed imbalance-aware objective is designed for.

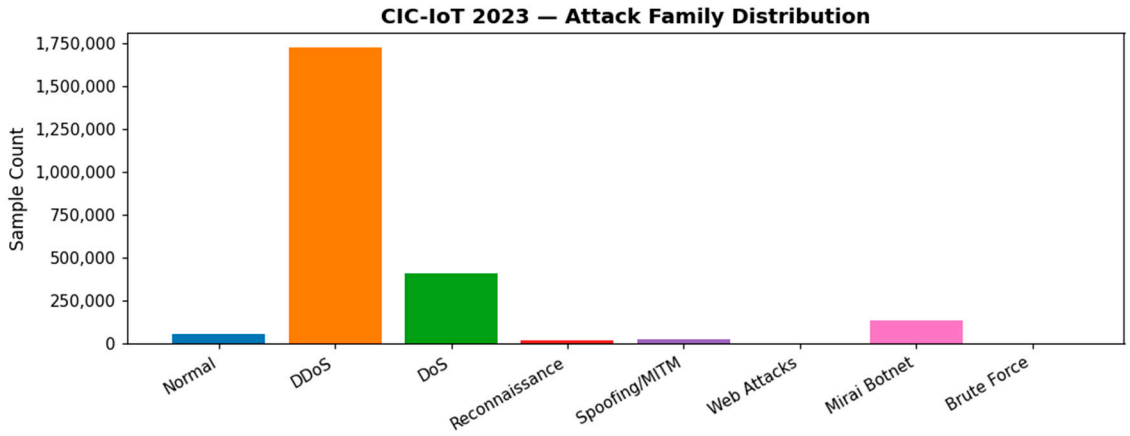


Figure 3. CIC-IoT2023 eight-class attack-family distribution.

CIC-IoT2023 is released as 169 CSV partitions; the experiments use the first 10 partitions in sorted filename order, which contain 2,366,956 flows, a subset adopted to keep training tractable on the available hardware. Table 1 reports the class distribution of this sample and of the training, validation and test partitions obtained with the stratified split described in Section 2.4.4; stratification preserves the class proportions in every partition. The largest family (DDoS) outnumbers the smallest (Brute Force) by 2627 to 1.

Table 1. Class distribution of the sample before and after the stratified 70/15/15 split.

Class	Full Sample	Train	Validation	Test
Normal	55,859	39,101	8379	8379
DDoS	1,723,607	1,206,525	258,541	258,541
DoS	409,683	286,778	61,452	61,453
Reconnaissance	17,985	12,590	2697	2698
Spoofing/MITM	24,760	17,332	3714	3714
Web Attacks	1186	830	178	178
Mirai Botnet	133,220	93,254	19,983	19,983
Brute Force	656	459	99	98
Total	**2,366,956**	**1,656,869**	**355,043**	**355,044**

2.4. Data Preprocessing

A four-stage preprocessing pipeline was applied to the dataset prior to training.

2.4.1. Label Mapping and Encoding

The per-sample fine-grained labels were mapped onto one of the eight target attack families, and the target labels were then transformed into a one-hot categorical representation suitable for the softmax classification head. Following the CIC-IoT2023 taxonomy [10], the Mirai Botnet family comprises the greeth flood, greip flood, and UDP-plain variants, whereas Spoofing/MITM comprises ARP and DNS spoofing only.

2.4.2. Handling of Missing and Invalid Values

Infinite values were first converted into undefined values and then treated as missing. Numerical attributes were imputed with the column mean of the training partition, and only numeric attributes were retained as model input, whereas non-numeric identifier columns were discarded.

2.4.3. Feature Scaling and Reshaping

All retained features were standardized so that each attribute has zero mean and unit variance. The imputation means and the standardisation parameters were estimated on the training partition only (Section 2.4.4) and then applied unchanged to the validation and test partitions, so that no statistic of the held-out data enters the preprocessing. CIC-IoT2023 provides 46 numeric flow descriptors and all of them were retained; that is, no feature selection or discarding was performed. The standardized vector was zero-padded with two zeros to a length of 48 so that it could be reshaped into the $8 \times 6 \times 1$ tensor required by the two branches [17,35]. This reshaping is a computational device rather than a claim that the descriptors possess spatial or temporal structure, but it does impose an ordering. The convolutional branch learns interactions only within fixed groups of neighbouring descriptors, and the bidirectional LSTM, like any recurrent network, is sensitive to the order of the eight six-dimensional segments it reads, even though it processes them in both directions. The native column order of CIC-IoT2023, identical for every flow, is therefore an architectural assumption of the model: it makes the input construction deterministic and reproducible, but we do not claim that it is optimal, and the sensitivity of the model to this order has not been assessed (Section 3.3.4). Because the feature count is below the target size, no truncation or column-order selection occurs, and the padding is fixed and identical for every sample. The resulting tensor simultaneously serves as the input of the convolutional and of the recurrent branch.

2.4.4. Stratified Splitting

A stratified 70/15/15 train–validation–test split was used in order to preserve the original class proportions of CIC-IoT2023 [9]. The split is performed on the raw data, before any preprocessing statistic is computed. Out of the 2,366,956 samples, 1,656,869 were used for training, 355,043 for validation, and 355,044 for testing. The validation set was used exclusively for early stopping and learning-rate scheduling, so that the reported test metrics were computed on data never seen during model selection.

2.5. Proposed Cross-Attention CNN-BiLSTM Model

The proposed framework, illustrated in Figure 4, processes the $8 \times 6 \times 1$ input tensor through two parallel branches followed by a bidirectional cross-attention fusion stage.

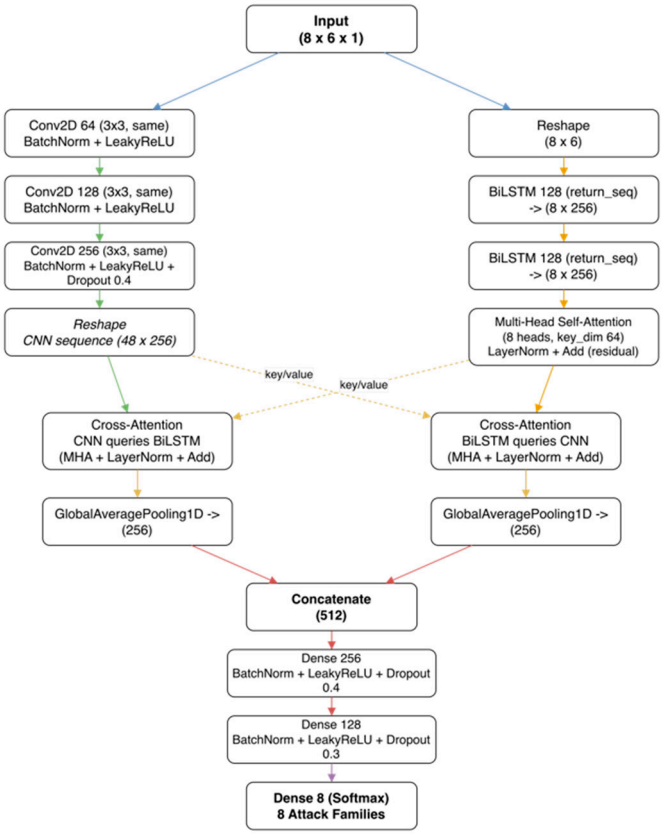


Figure 4. Proposed cross-attention CNN-BiLSTM architecture.

The convolutional branch consists of three convolutional blocks with 3×3 kernels and same padding, holding 64, 128, and 256 filters respectively, each followed by batch normalization [36] and a LeakyReLU activation with a negative slope of 0.1. A dropout layer with a rate of 0.4 is applied after the third block. In order to preserve the spatial structure rather than collapsing it into a single vector, no max-pooling is used and the feature map is instead reshaped into a sequence of 48 spatial positions of dimension 256, which serves as input of the cross-attention block.

The sequential branch processes the same input as a sequence of eight six-dimensional feature segments through two stacked bidirectional LSTM layers, each with 128 hidden units per direction [15,17]. The stacked bidirectional encoding produces a sequence of eight 256-dimensional steps that captures dependencies among the feature segments read in both the forward and the backward direction. A multi-head self-attention block [30] with eight heads and a key dimension of 64 is then applied to refine the sequential representation, followed by layer normalization [32] and a residual connection.

The bidirectional cross-attention block of (2) and (3), also using eight heads and a key dimension of 64, lets each branch attend to the complementary modality. Each attended sequence is reduced by global average pooling to a 256-dimensional vector, and the two pooled vectors are concatenated into a 512-dimensional fusion vector. This vector is fed into a multi-layer perceptron of two dense layers of 256 and 128 units, interleaved with batch normalization [36], LeakyReLU activation, and dropout [37] with rates of 0.4 and 0.3. The final softmax layer outputs the probability distribution over the eight attack families.

2.6. Experimental Setup

All experiments were implemented in Python 3.12 with the Keras deep learning library running on the TensorFlow 2.20 backend [38], and the training and the evaluation were carried out on a laptop equipped with an eight-core Apple M1 processor and 16 GB of

RAM. The model was optimized with Adam [39] with an initial learning rate of 0.001, using the focal loss of (6). Class-balanced weights derived from the training distribution were applied at every batch. Training ran for up to 60 epochs with a batch size of 128, the learning rate being reduced by a factor of five on validation-loss plateaus, and early stopping monitored the validation accuracy with a patience of ten epochs.

2.7. Evaluation Criteria

Four standard metrics were adopted for the multi-class classification task, namely accuracy, precision, recall, and F1-score [4,7], defined in (8)–(11), where TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. Per-class metrics are reported together with the macro-averaged precision, recall, and F1-score, and a one-versus-rest Receiver Operating Characteristic (ROC) analysis is performed for each attack family.

$$\mathrm{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$\mathrm{Prec} = \frac{TP}{TP + FP} \tag{9}$$

$$\mathrm{Rec} = \frac{TP}{TP + FN} \tag{10}$$

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Prec} \cdot \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}} \tag{11}$$

3. Results and Discussion

3.1. Results on CIC-IoT2023

Table 2 reports the per-class precision, recall, and F1-score of the proposed model on CIC-IoT2023. On this large-scale benchmark of more than 3.5×10^5 test flows, the model reached a global accuracy of 99.06% and a weighted F1-score of 98.99%, with a near-perfect detection of the dominant volumetric families, since DDoS, DoS, and Mirai Botnet all exceed an F1-score of 0.998. The macro F1-score of 68.57% is markedly lower than the weighted value because two extremely rare families, Web Attacks with 178 samples, that is 0.05% of the test set, and Brute Force with 98 samples, that is 0.028%, are still poorly recognized, and the Reconnaissance and Spoofing/MITM families reach recalls of only 62.5% and 60.4%.

Table 2. Per-class performance of the proposed model on CIC-IoT2023, with the per-class Matthews correlation coefficient (MCC) and the 95% Wilson confidence interval of recall.

Class	Precision	Recall	F1-Score	Support	MCC	Recall 95% CI
Normal	0.7917	0.9599	0.8677	8379	0.8684	[0.9555, 0.9639]
DDoS	0.9991	0.9998	0.9994	258,541	0.9980	[0.9997, 0.9998]
DoS	0.9997	0.9970	0.9983	61,453	0.9980	[0.9965, 0.9974]
Reconnaissance	0.7483	0.6249	0.6811	2698	0.6817	[0.6065, 0.6430]
Spoofing/MITM	0.8480	0.6037	0.7053	3714	0.7130	[0.5878, 0.6193]
Web Attacks	0.0000	0.0000	0.0000	178	0.0000	[0.0000, 0.0211]
Mirai Botnet	0.9999	0.9994	0.9997	19,983	0.9997	[0.9990, 0.9997]
Brute Force	1.0000	0.1327	0.2342	98	0.3642	[0.0792, 0.2138]
Macro average	0.7983	0.6647	0.6857	355,044	—	—
Weighted average	0.9904	0.9906	0.9899	355,044	—	—

Figure 5 shows that the misclassifications are concentrated in the rarest and smallest families: the Web Attacks flows are spread over Spoofing/MITM (40%), Normal (30%), and Reconnaissance (30%), 42% of the Brute Force flows are absorbed by the Normal class, and about one third of the Reconnaissance and Spoofing/MITM flows are also labelled as Normal, which is consistent with the low recalls reported in Table 2.

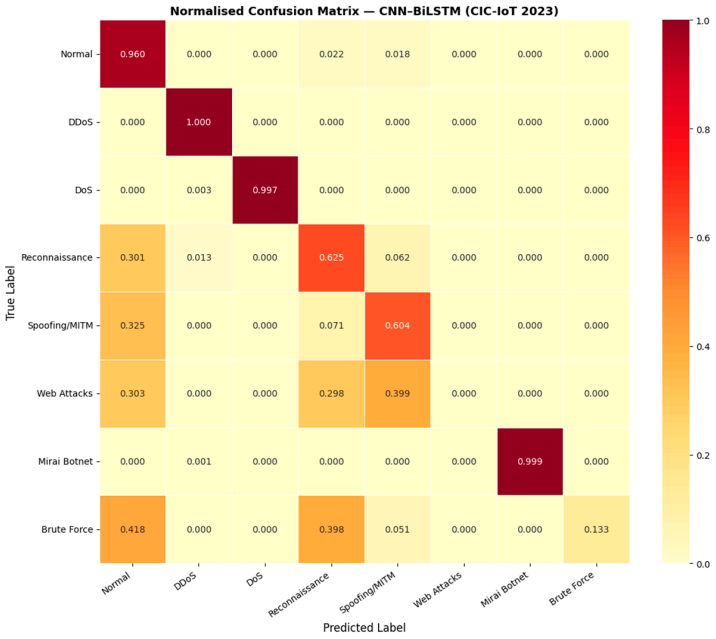


Figure 5. Row-normalized confusion matrix of the proposed model on CIC-IoT2023.

Interestingly, the one-versus-rest ROC analysis of Figure 6 yields a high Area Under the Curve (AUC) for every family, including Web Attacks with 0.9963 and Brute Force with 0.9965, even though their F1-scores computed at the arg max decision rule are low. Combined with the very low recall of these families at the arg max rule, under which the Web Attacks class is never predicted and Brute Force only 13 times, this pattern is consistent with a decision-threshold effect under extreme class imbalance rather than a lack of discriminative features. ROC-AUC is, however, known to be optimistic at such low prevalence, and confirming this interpretation requires precision–recall and threshold analysis, which is left to future work; threshold calibration or targeted resampling is therefore a plausible, but not yet verified, remedy.

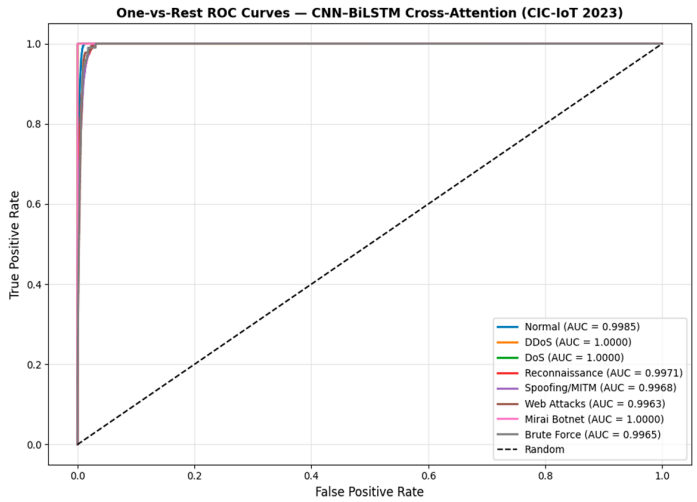


Figure 6. One-versus-rest ROC curves of the proposed model on CIC-IoT2023.

The training and validation curves of Figure 7 confirm a stable convergence with a negligible train–validation gap, which indicates that the proposed model generalizes well on the CIC-IoT2023 distribution without overfitting. Training stopped early at epoch 34, and the weights of epoch 24, which achieved the highest validation accuracy, were restored.

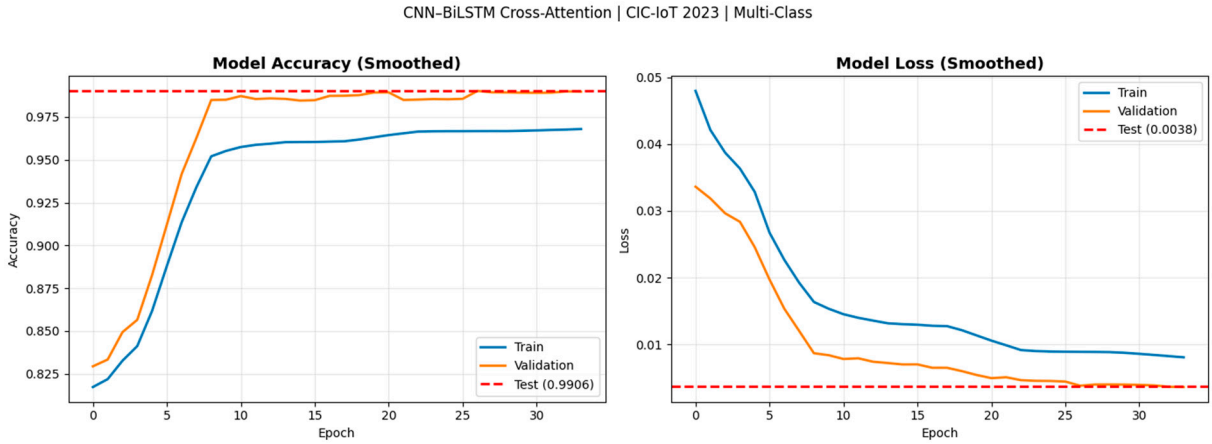


Figure 7. Training and validation accuracy and loss curves of the proposed model on CIC-IoT2023.

Because accuracy and weighted F1 are dominated by the majority families, Table 3 complements them with imbalance-sensitive metrics computed from the confusion matrix of the test partition. While accuracy reaches 99.06% (95% Wilson confidence interval [99.03%, 99.09%]), the balanced accuracy is only 66.47%, and the geometric mean of the per-class recalls is zero because no Web Attack flow is recognized. The multi-class MCC and Cohen's κ remain high (0.9785 and 0.9784) because they are also dominated by the majority families; the per-class MCC of Table 2, 0.00 for Web Attacks and 0.36 for Brute Force, gives the faithful picture. The model detects none of the 178 Web Attack flows (recall 0%, 95% CI [0%, 2.1%]) and 13 of the 98 Brute Force flows (recall 13.3%, 95% CI [7.9%, 21.4%]); it never predicts the Web Attacks class, and each of its 13 Brute Force predictions is correct. Moreover, about 70% of all misclassified attack flows (2116 of 3001) are assigned to the Normal class, which is the operationally most costly type of error. A second, independent training run of the same pipeline yielded 99.02% accuracy and a 68.54% macro F1-score, which indicates that these figures are stable with respect to random initialisation; the confidence intervals above nevertheless reflect only the sampling uncertainty of the test partition, and two runs do not constitute a formal multi-seed study.

Table 3. Aggregate and imbalance-sensitive metrics of the proposed model on the test partition.

Metric	Value
Accuracy (95% CI)	99.06% [99.03, 99.09]
Weighted F1-score	98.99%
Macro F1-score	68.57%
Balanced accuracy	66.47%
G-mean of per-class recalls	0.00%
Matthews correlation coefficient	0.9785
Cohen's κ	0.9784

Table 4 compares the proposed model with four deep learning baselines, namely a standalone CNN, a standalone LSTM, a hybrid CNN-LSTM without attention, and a hybrid CNN-BiLSTM without attention. In order to ensure a fair comparison that isolates

the effect of the architecture, all baselines were trained with the same focal loss, class-balanced weighting, optimizer, and training budget as the proposed model, and they were retrained on exactly the same corrected pipeline: the same ten partitions, label mapping, training-only preprocessing and stratified 70/15/15 split, and the same test partition of 355,044 flows. Every learned baseline comfortably exceeds the 72.82% majority-class floor, that is the trivial prior of always predicting the dominant DDoS family, and the strongest baseline, CNN-BiLSTM without attention, reaches 90.14% accuracy and a 64.19% macro F1-score, against 99.06% and 68.57% for the proposed model. The CNN-BiLSTM attains a higher macro precision (87.65% against 79.83%) because its single Web Attacks prediction is correct, whereas the proposed model never predicts that class; the proposed model is ahead on accuracy, macro recall, macro F1-score and weighted F1-score.

Table 4. Multi-class performance and number of parameters of the proposed model against deep learning baselines on CIC-IoT2023.

Model	Acc.	Macro P	Macro R	Macro F1	Wtd. F1	Params
LSTM	88.05%	75.12%	59.34%	62.09%	87.01%	268,168
CNN-LSTM	89.37%	75.87%	60.30%	63.38%	88.69%	552,584
CNN	89.84%	75.47%	59.19%	62.56%	88.99%	87,176
CNN-BiLSTM	90.14%	87.65%	62.08%	64.19%	90.10%	1,028,744
Proposed model	**99.06%**	**79.83%**	**66.47%**	**68.57%**	**98.99%**	2,650,504

Table 4 also lists the number of parameters of each model. The proposed model contains 2,650,504 parameters (10.6 MB in 32-bit precision), about 2.6 times more than the strongest baseline (CNN-BiLSTM, 1,028,744 parameters). Measured on an Apple M1 Pro processor without GPU acceleration, it classifies about 1190 flows per second in batched inference (0.84 ms per flow), with a median single-flow latency of 5.7 ms, which is compatible with inspection at an edge gateway; its larger size nevertheless matters for deployment on constrained IoT devices.

3.2. Positioning with Respect to Recent Studies

All the studies listed in Table 5 were evaluated on the same CIC-IoT2023 benchmark [10], which is a necessary but not a sufficient condition for a direct comparison; a fair reading, however, must account for the classification task and the label granularity of each study, since both strongly affect the reported accuracy. Table 5 therefore reports, for every compared model, the technique used and the task with its class granularity. Binary normal-versus-attack detection is an intrinsically easier problem than fine-grained multi-class attack identification, so the two settings must be read separately rather than on a single accuracy scale. In the binary setting, the federated DNN of [40], the CNN-LSTM of [41], and the CNN of [42] reach 98.42–99.00%; these results cannot be compared directly with the eight-class problem addressed here.

Beyond these general-purpose binary detectors, the reported accuracies depend strongly on the class granularity and on the evaluation protocol. The genetically optimized gradient boosting of [43] reaches 95.00% in distinguishing web attacks from benign traffic, and the multi-layer-perceptron intrusion detection and prevention system of [44] reaches 85.00% on a six-class task comprising benign traffic and five attack families; neither task is comparable with the eight-family problem addressed here. Three studies report results for that task on CIC-IoT2023: the transferable deep-reinforcement-learning NIDS of [45] reaches 95.60%, although on class-balanced samples of more than 10,000 flows rather than on the native distribution; the optimized gradient-boosting framework of [46]

attains 99.20% (CatBoost), 99.38% (LightGBM) and 99.59% (XGBoost); and the CNN of [47] attains 99.02%.

On the eight-class task, the proposed model, with 99.06%, lies within the range of the results of [46] and [47] but does not exceed them. These figures are, moreover, obtained under different protocols: [46] uses a working set of 5,000,000 flows with stratified undersampling that retains at least 1000 samples per class, selects 23 of the 46 features and applies an 80/20 split; [47] uses 4,668,653 flows (10% of the dataset), the 20 most important features according to a random-forest model and a stratified 80/20 split; and the present study uses 2,366,956 flows from 10 of the 169 partitions with their native class imbalance, all 46 features and a stratified 70/15/15 split. Because sampling, class balancing and feature selection differ, accuracy alone does not support a claim of superiority in either direction, and Table 5 positions the proposed model rather than ranking it. Its distinguishing characteristics are methodological: it is a fully end-to-end deep architecture trained on the native, non-resampled class distribution with an imbalance-aware focal loss, accompanied by a per-class and imbalance-sensitive analysis. Hybrid convolutional-recurrent and attention-based designs have likewise proven effective for IoT intrusion detection on other benchmarks: a hybrid CNN-LSTM model was analysed for robustness across the ToN_IoT and UNSW-NB15 datasets [48], deep residual networks with attention mechanisms were applied to IoT intrusion detection [49], and machine-learning-driven algorithms were assessed for discriminating DDoS traffic within IoT systems [50].

Two factors should be kept in mind when interpreting these figures. First, label granularity: studies that group the traffic into fewer or coarser categories face an easier task, while the proposed model preserves an operationally meaningful eight-family taxonomy. Second, the evaluation protocol: the present results are obtained on the native, non-resampled test distribution of CIC-IoT2023, which is dominated by benign and volumetric traffic, so the reported accuracy reflects realistic operating conditions rather than an artificially balanced test set. Consistent with these results, hybrid convolutional-recurrent architectures have also proven effective for IoT intrusion detection on other benchmarks, including a lightweight CNN-LSTM model evaluated on UNSW-NB15, a refined LSTM detector for denial-of-service attacks on NSL-KDD, and a spatio-temporal CNN-TCN-CapsuleNet-LSTM fusion network.

Table 5. Comparison with recent intrusion detection studies on CIC-IoT2023.

Ref.	Year	Method	Task	Acc. (%)
[43]	2023	Gradient boosting	2-class	95.00
[41]	2024	CNN-LSTM	2-class	98.42
[40]	2023	DNN (federated learning)	2-class	99.00
[42]	2025	CNN	2-class	99.00
[44]	2025	Multi-layer perceptron	6-class	85.00
[46]	2026	CatBoost	8-class	99.20
[46]	2026	LightGBM	8-class	99.38
[46]	2026	XGBoost	8-class	99.59
[45]	2024	Deep reinforcement learning	8-class	95.60
[47]	2026	CNN	8-class	99.02
This work	**2026**	**Cross-attention CNN-BiLSTM**	**8-class**	**99.06**

3.3. Discussion

3.3.1. Ablation Insight

The proposed model reaches 99.06% accuracy and a 68.57% macro F1-score, against 90.14% and 64.19% for the attention-free CNN-BiLSTM retrained on the same pipeline; the

95% confidence intervals of the two accuracies do not overlap ([99.03%, 99.09%] against [90.04%, 90.24%]). This difference cannot be attributed to any particular architectural component. The two models differ simultaneously in the presence of multi-head self-attention, in the bidirectional cross-attention between the branches, in the design of their convolutional and recurrent backbones, and in capacity (2,650,504 versus 1,028,744 parameters). The comparison therefore indicates only that the proposed configuration as a whole performs better than the baselines; it does not show that the attention-based fusion, cross-attention or self-attention is responsible for the difference, nor that the larger capacity is not. Isolating these factors requires a component-wise ablation with a capacity-matched control, which we identify as the primary limitation of this study. For the same reason, the difference between the CNN-BiLSTM and CNN-LSTM baselines (90.14% against 89.37%) should be read as a comparison of complete configurations rather than as an isolated effect of bidirectional encoding.

3.3.2. Effect of the Focal Loss and of the Class Weighting

All models of this study, including the baselines, were trained with the focal loss and class-balanced weighting, so that the comparison reflects architectural differences rather than the training objective. This imbalance-aware objective steers the proposed model to a macro precision of 79.83%. This average is lowered by the Web Attacks class, which is never predicted, whereas the precision exceeds 0.99 for DDoS, DoS, Mirai Botnet, and Brute Force, so that when the model assigns a flow to one of these families it is rarely wrong. For security operations, such a high precision translates into fewer false alarms [29].

3.3.3. Impact of the Extreme Class Imbalance

The gap between the weighted F1-score of 98.99% and the macro F1-score of 68.57% is largely caused by the rarest families. Web Attacks and Brute Force account for only 178 and 98 of the more than 3.5×10^5 test flows, that is 0.05% and 0.028% of the data, respectively. At this sampling rate, even a focal loss and class-balanced weighting cannot supply enough discriminative signal: the model never predicts the Web Attacks class, which yields an F1-score of 0 for this family, and it recovers only 13.3% of the Brute Force flows (F1-score of 23.42%), although each of its 13 Brute Force predictions is correct. The Spoofing/MITM and Reconnaissance families, with recalls of 60.4% and 62.5%, also contribute to the gap, since about one third of their flows are labelled as Normal. This behavior is consistent with a decision-threshold effect under extreme imbalance, as suggested by the high one-versus-rest AUC of these families (Section 3.1); however, ROC-AUC is optimistic at such low prevalence, and confirming this interpretation requires precision–recall and threshold analysis, which we leave to future work. For the same reason, the accuracy and the weighted F1-score must always be read together with the macro-averaged and imbalance-sensitive metrics of Table 3, which expose the minority-class behavior that the aggregate figures conceal.

3.3.4. Limitations

This study has several limitations. First, the model recognizes none of the Web Attack flows and only 13.3% of the Brute Force flows, which jointly represent less than 0.08% of the test set, and about 70% of the misclassified attack flows are labelled as Normal traffic; reliably detecting such ultra-minority classes will likely require dedicated minority over-sampling, threshold calibration, or few-shot detection strategies rather than loss-level re-weighting alone [4]. Second, the proposed model differs from the attention-free baseline in its attention mechanisms, backbone and capacity, so the individual contributions of self-attention, cross-attention and model size are not isolated; a component-wise ablation with a capacity-matched control is required. Third, the interpretation of the rare-class

errors as a decision-threshold effect is not yet confirmed by precision–recall or threshold analysis. Fourth, the experiments use 10 of the 169 CIC-IoT2023 partitions and a single benchmark, and two independent training runs, although consistent (99.06% and 99.02% accuracy), do not replace a formal multi-seed study, so neither seed-to-seed variability nor generalisation to other traffic distributions is fully established. Finally, the native order of the flow descriptors is an architectural assumption of the model, and the sensitivity of the model to this order has not been assessed.

4. Conclusions

The fine-grained identification of the specific attack family, and not merely the separation of normal from malicious traffic, is what enables analysts and automated defenses to react in a targeted manner, yet it remains difficult on modern IoT datasets because of severe class imbalance and the overlap of feature distributions across attack families. To address this gap, this paper proposed an imbalance-aware cross-attention CNN-BiLSTM framework that fuses a convolutional branch and a bidirectional-LSTM branch through a bidirectional cross-attention mechanism, and that is trained with a focal loss combined with class-balanced weighting.

On the CIC-IoT2023 benchmark, the proposed model reached 99.06% accuracy and a 98.99% weighted F1-score, outperforming standalone CNN, standalone LSTM, CNN-LSTM, and CNN-BiLSTM baselines retrained on the same corrected data pipeline under an identical objective and budget. Its accuracy lies within the range reported by recent eight-family studies on the same dataset, which, however, use different sampling, balancing and feature-selection protocols, so no claim of superiority over them is made.

These aggregate figures must, however, be read together with the macro-averaged results. The macro F1-score is 68.57% and the balanced accuracy 66.47%, and the geometric mean of the per-class recalls is zero because the model recognizes none of the Web Attack flows (F1 = 0). It identifies only 13.3% of the Brute Force flows (F1 = 23.42%) and 60.4% of the Spoofing/MITM flows, and about 70% of the misclassified attack flows are labelled as Normal traffic. The proposed framework is therefore reliable for the high-volume families, but it cannot be regarded as a dependable detector of the rarest attacks.

The main contribution of this work lies in the attention-based fusion of the convolutional and the bidirectional-recurrent representations, which combines multi-head self-attention with bidirectional cross-attention. The proposed model reaches 99.06% accuracy against 90.14% for the attention-free CNN-BiLSTM baseline; because the two models differ in their attention mechanisms, backbone and capacity, the present experiments do not establish which component is responsible for this difference. Unlike the tree-ensemble and single-branch deep models previously reported on CIC-IoT2023, the proposed approach is a fully end-to-end deep architecture that couples imbalance-aware training with a per-class, imbalance-sensitive and ROC-based analysis. The high one-versus-rest AUC of the rarest families (above 0.996), combined with their very low recall at the arg max rule, is consistent with a decision-threshold effect rather than a representational failure, although confirming this interpretation requires precision–recall and threshold analysis.

Future work will prioritize a component-wise ablation separating self-attention, unidirectional and bidirectional cross-attention, and model capacity; precision–recall and threshold analysis for the rarest families, together with threshold calibration and targeted minority over-sampling; a multi-seed evaluation with formal significance testing; a feature-order sensitivity study; and validation on additional datasets. A hierarchical detect-then-classify head, finer-grained attack taxonomies, and lightweight deployment-oriented variants will also be investigated.

https://doi.org/10.3390/fi18100521

Figure 8 summarises the operational value of fine-grained attack classification for security analysts.

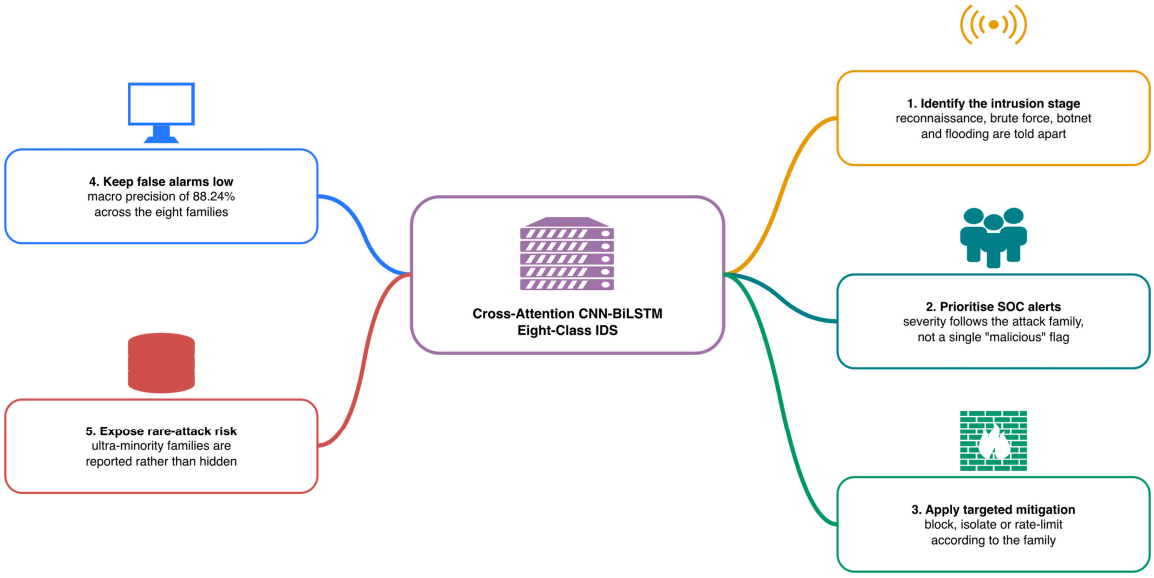


Figure 8. Operational value of fine-grained attack classification. Binary detection establishes only whether traffic is malicious, whereas identifying the specific attack family indicates the stage of the intrusion and the appropriate response.

Author Contributions: Conceptualization, M.A.F.; methodology, M.A.F. and A.N.; software, M.A.F. and A.N.; validation, A.N. and N.E.I.; formal analysis, R.B.S.; investigation, M.A.F. and A.N.; resources, M.A.F. and A.N.; data curation, M.A.F., R.B.S. and N.E.I.; writing—original draft preparation, M.A.F. and N.E.I.; writing—review and editing, R.B.S.; visualization, M.A.F. and R.B.S.; supervision, M.A.F. and A.N.; project administration, A.N., R.B.S. and N.E.I.; funding acquisition, N.E.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The CIC-IoT2023 dataset used in this study is publicly available from the Canadian Institute for Cybersecurity and is described in [10]. No new data were created in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liao, H.; Murah, M.Z.; Hasan, M.K.A.; Aman, A.H.M.; Fang, J.; Hu, X.; Khan, A.H. A survey of deep learning technologies for intrusion detection in Internet of Things. *IEEE Access* **2024**, *12*, 4745–4761. [CrossRef]
2. Gelgi, M.; Guan, Y.; Arunachala, S.; Samba Siva Rao, M.; Dragoni, N. Systematic literature review of IoT botnet DDoS attacks and evaluation of detection techniques. *Sensors* **2024**, *24*, 3571. [CrossRef] [PubMed]
3. Awajan, A. A novel deep learning-based intrusion detection system for IoT networks. *Computers* **2023**, *12*, 34. [CrossRef]
4. Morshedi, H.; Abawajy, J.; Fotouhi, F. A comprehensive review of deep learning techniques for anomaly detection in IoT networks: Methods, challenges, and datasets. *Eng. Rep.* **2025**, *7*, e70113. [CrossRef]
5. Ahmad, I.; Basheri, M.; Iqbal, M.J.; Rahim, A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access* **2018**, *6*, 33789–33795. [CrossRef]
6. Tama, B.A.; Comuzzi, M.; Rhee, K.-H. TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. *IEEE Access* **2019**, *7*, 94497–94507. [CrossRef]
7. Maseer, Z.K.; Yusof, R.; Bahaman, N.; Mostafa, S.A.; Foozy, C.F.M. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access* **2021**, *9*, 22351–22370. [CrossRef]
8. Disha, R.A.; Waheed, S. Performance analysis of machine learning models for intrusion detection system using Gini impurity-based weighted random forest (GIWRF) feature selection technique. *Cybersecurity* **2022**, *5*, 1. [CrossRef]

9. Hozouri, A.; Mirzaei, A.; Effatparvar, M. A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. *Discov. Artif. Intell.* **2025**, *5*, 314. [CrossRef]
10. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors* **2023**, *23*, 5941. [CrossRef] [PubMed]
11. Wu, K.; Chen, Z.; Li, W. A novel intrusion detection model for a massive network using convolutional neural networks. *IEEE Access* **2018**, *6*, 50850–50859. [CrossRef]
12. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep learning approach for intelligent intrusion detection system. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
13. Al-Turaiki, I.; Altwaijry, N. A convolutional neural network for improved anomaly-based network intrusion detection. *Big Data* **2021**, *9*, 233–252. [CrossRef] [PubMed]
14. Yin, C.; Zhu, Y.; Fei, J.; He, X. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
15. Imrana, Y.; Xiang, Y.; Ali, L.; Abdul-Rauf, Z. A bidirectional LSTM deep learning approach for intrusion detection. *Expert Syst. Appl.* **2021**, *185*, 115524. [CrossRef]
16. Kasongo, S.M. A deep learning technique for intrusion detection system using a recurrent neural networks based framework. *Comput. Commun.* **2023**, *199*, 113–125. [CrossRef]
17. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
18. Sun, P.; Liu, P.; Li, Q.; Liu, C.; Lu, X.; Hao, R.; Chen, J. DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. *Secur. Commun. Netw.* **2020**, *2020*, 8890306. [CrossRef]
19. Abdallah, M.; Le-Khac, N.-A.; Jahromi, H.; Jurcut, A.D. A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs. In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; Art. no. 79. [CrossRef]
20. Halbouni, A.; Gunawan, T.S.; Habaebi, M.H.; Halbouni, M.; Kartiwi, M.; Ahmad, R. CNN-LSTM: Hybrid deep neural network for network intrusion detection system. *IEEE Access* **2022**, *10*, 99837–99849. [CrossRef]
21. Said, R.B.; Sabir, Z.; Askerzade, I. CNN-BiLSTM: A hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection. *IEEE Access* **2023**, *11*, 138732–138747. [CrossRef]
22. Altunay, H.C.; Albayrak, Z. A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. *Eng. Sci. Technol. Int. J.* **2023**, *38*, 101322. [CrossRef]
23. Bamber, S.S.; Katkuri, A.V.R.; Sharma, S.; Angurala, M. A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system. *Comput. Secur.* **2025**, *148*, 104146. [CrossRef]
24. Laghrissi, F.; Douzi, S.; Douzi, K.; Hssina, B. IDS-attention: An efficient algorithm for intrusion detection systems using attention mechanism. *J. Big Data* **2021**, *8*, 149. [CrossRef]
25. Wu, Z.; Zhang, H.; Wang, P.; Sun, Z. RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE Access* **2022**, *10*, 64375–64387. [CrossRef]
26. Xu, H.; Sun, L.; Fan, G.; Li, W.; Kuang, G. A hierarchical intrusion detection model combining multiple deep learning models with attention mechanism. *IEEE Access* **2023**, *11*, 66212–66226. [CrossRef]
27. Long, Z.; Yan, H.; Shen, G.; Zhang, X.; He, H.; Cheng, L. A transformer-based network intrusion detection approach for cloud security. *J. Cloud Comput.* **2024**, *13*, 5. [CrossRef]
28. Al-Omar, B.; Trabelsi, Z. Intrusion detection using attention-based CNN-LSTM model. In *Proceedings of the Artificial Intelligence Applications and Innovations, Leon, Spain, 14–17 June 2023*; Springer: Cham, Switzerland, 2023; Volume 675, pp. 515–526. [CrossRef]
29. Alashjaee, A.M. Deep learning for network security: An attention-CNN-LSTM model for accurate intrusion detection. *Sci. Rep.* **2025**, *15*, 21856. [CrossRef] [PubMed]
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 6000–6010.
31. Wang, J.; Yu, L.; Tian, S. Cross-attention interaction learning network for multi-model image fusion via transformer. *Eng. Appl. Artif. Intell.* **2025**, *139*, 109583. [CrossRef]
32. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450. [CrossRef]
33. Tseng, S.-M.; Wang, Y.-Q.; Wang, Y.-C. Multi-class intrusion detection based on transformer for IoT networks using CIC-IoT-2023 dataset. *Future Internet* **2024**, *16*, 284. [CrossRef]
34. Jony, A.I.; Arnob, A.K.B. A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset. *J. Edge Comput.* **2024**, *3*, 28–42. [CrossRef]
35. Ullah, I.; Mahmoud, Q.H. Design and development of a deep learning-based model for anomaly detection in IoT networks. *IEEE Access* **2021**, *9*, 103906–103926. [CrossRef]

36. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
37. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfiting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
38. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
40. Abbas, S.; Al Hejaili, A.; Sampedro, G.A.; Abisado, M.; Almadhor, A.S.; Shahzad, T.; Ouahada, K. A novel federated edge learning approach for detecting cyberattacks in IoT infrastructures. *IEEE Access* **2023**, *11*, 112189–112198. [CrossRef]
41. Gueriani, A.; Kheddar, H.; Mazari, A.C. Enhancing IoT security with CNN and LSTM-based intrusion detection systems. In Proceedings of the 6th International Conference on Pattern Analysis and Intelligent Systems, El Oued, Algeria, 24–25 April 2024; pp. 1–7. [CrossRef]
42. Wahab, S.A.; Sultana, S.; Tariq, N.; Mujahid, M.; Khan, J.A.; Mylonas, A. A multi-class intrusion detection system for DDoS attacks in IoT networks using deep learning and transformers. *Sensors* **2025**, *25*, 4845. [CrossRef] [PubMed]
43. Jaradat, A.S.; Nasayreh, A.; Al-Na'amneh, Q.; Gharaibeh, H.; Al Mamlook, R.E. Genetic optimization techniques for enhancing web attacks classification in machine learning. In Proceedings of the IEEE International Conference on Dependable, Autonomic and Secure Computing, Abu Dhabi, United Arab Emirates, 14–17 November 2023; pp. 130–136. [CrossRef]
44. Erskine, S.K. Real-time large-scale intrusion detection and prevention system (IDPS) CICIoT dataset traffic assessment based on deep learning. *Appl. Syst. Innov.* **2025**, *8*, 52. [CrossRef]
45. He, M.; Wang, X.; Wei, P.; Yang, L.; Teng, Y.; Lyu, R. Reinforcement learning meets network intrusion detection: A transferable and adaptable framework for anomaly behavior identification. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 2477–2492. [CrossRef]
46. Almahaqeri, S.A.; Almourish, M.H.; Nasser, A.A.; Alghawli, A.S.A.; Elsayed, A.A.K.; Alhejoj, A.N. An optimized gradientboosting framework for IoT intrusion detection: A comprehensive evaluation on the CICIoT2023 dataset. *Sci. Rep.* **2026**, *16*, 16909. [CrossRef] [PubMed]
47. Andrade, I.; Mahadik, S.S.; Mukherjee, M.; Pawar, P.M.; Muthalagu, R. Deep learning based intelligent intrusion detection sys-tem for large-scale IoT networks. *arXiv* **2026**, arXiv:2603.16342. [CrossRef]
48. Othman, A.M.H.; Ab Razak, M.F.; Firdaus, A.; Tahaei, H.; Gheisari, M. A hybrid CNN–LSTM model for IoT intrusion detection: A robustness analysis across datasets. *Future Internet* **2026**, *18*, 345. [CrossRef]
49. Cui, B.; Chai, Y.; Yang, Z.; Li, K. Intrusion detection in IoT using deep residual networks with attention mechanisms. *Future Internet* **2024**, *16*, 255. [CrossRef]
50. Alenezi, M.N. Significance of machine learning-driven algorithms for effective discrimination of DDoS traffic within IoT systems. *Future Internet* **2025**, *17*, 266. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.