

Article

Explainable Hybrid CNN–XGBoost Framework for Multi-Class IoT Intrusion Detection with Leakage-Aware Feature Selection

Deemah AlFuraih , Lotfi Mhamdi  and Abdullah S. Karar * 

College of Computer and Systems Engineering, Abdullah Al Salem University, Khaldiya, Kuwait; s2242100009@aasu.edu.kw (D.A.); lotfi.mhamdi@aasu.edu.kw (L.M.)

* Correspondence: abdullah.karar@aasu.edu.kw

Abstract

The rapid deployment of Internet of Things (IoT) devices has increased exposure to a diverse array of evolving cyberattacks, motivating the need for accurate and interpretable intrusion detection systems (IDS). In this work, we develop an explainable hybrid Convolutional Neural Network–Extreme Gradient Boosting (CNN–XGBoost) framework for multi-class IoT attack classification using the CIC IoT-DIAD 2024 dataset. Network-traffic records are preprocessed and standardized using a scalable, chunk-wise workflow, after which a compact top-k subset of features is selected via Random Forest importance ranking. To reduce selection bias, a leakage-prone feature-ranking strategy is compared with a leakage-aware strategy in which features are ranked using only the training data within each split. Subsequently, a one-dimensional Convolutional Neural Network (CNN) learns a 128-dimensional representation from the selected predictors, and XGBoost performs the final multi-class classification. Under the leakage-aware protocol, the proposed model achieves 0.9324 accuracy with 0.5910 macro-F1. Results indicate that leakage-aware selection provides a more defensible estimate of generalization while maintaining competitive detection performance. Finally, SHapley Additive exPlanations (SHAP) is used to interpret the model’s decisions in the learned latent space. The analysis shows that only a small number of embedding dimensions contribute most of the decision evidence, which can aid analyst triage, although the explanations remain indirect with respect to the original traffic features.

Keywords: IoT intrusion detection; multi-class attack classification; hybrid deep learning; CNN feature extraction; XGBoost; leakage-aware feature selection; explainable AI; SHAP; SHAP interactions; network traffic security



Academic Editor: Andrzej Białas

Received: 26 January 2026

Revised: 23 February 2026

Accepted: 24 February 2026

Published: 26 February 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the rapid growth of connected systems and AI-enabled services, cyber attacks have become an increasingly persistent threat to individuals and organizations. AI-driven intrusion detection has therefore attracted substantial attention, since learning-based models can improve attack prediction and reduce potential damage when compared to purely manual or rule-based approaches [1]. At the same time, security operators and stakeholders often require transparent decision rationales to build trust in black-box AI systems, motivating the integration of explainable AI (XAI) into cybersecurity pipelines [2].

To contextualize this trend, our review of publications indexed on ScienceDirect indicates a sustained increase in research output at the intersection of cybersecurity, explainable AI, and IoT security. Figure 1 summarizes the publication growth from 2002 to 2025 for

cybersecurity in general, as well as the smaller (but increasing) body of work that jointly considers cybersecurity with XAI and IoT.

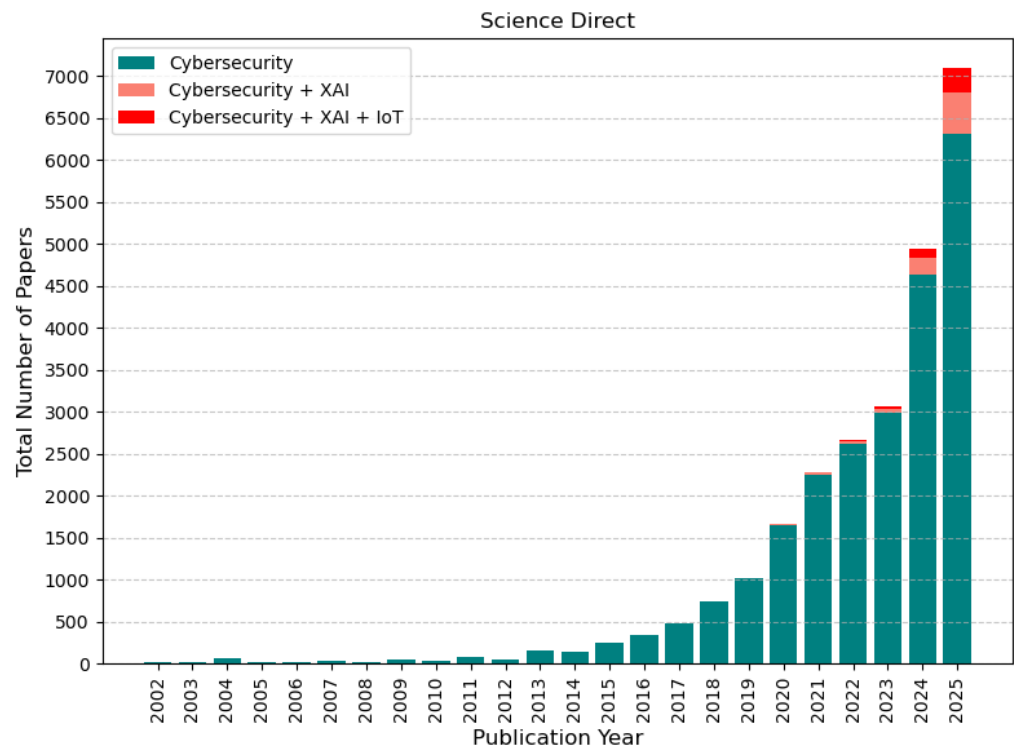


Figure 1. ScienceDirect publications trend (2002–2025) for Cybersecurity, Cybersecurity–XAI, and Cybersecurity–XAI–IoT.

Intrusion detection systems (IDSs) are widely used to monitor network behavior and raise alarms upon detecting abnormal or malicious activity. IDS methods are commonly categorized into signature-based and anomaly-based systems [3]. Signature-based IDS compare traffic against known attack patterns (signatures), offering effective detection of known threats but requiring frequent updates and typically failing against unseen attacks. In contrast, anomaly-based IDS learn profiles of normal behavior and can therefore detect novel or evolving attacks, although they may suffer from higher false positives under non-stationary traffic conditions [4]. Because IoT and wireless environments are exposed to diverse threats, such as denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks and malware variants, hybrid IDS designs that combine signature-based filtering with anomaly-based learning are often preferable [5,6].

Recent work highlights the importance of robust data collection, feature engineering, and realistic evaluation. For example, high-speed packet capture considerations have been studied in software-based capture architectures [7–9], while deployment considerations such as IDS sensor placement have also been investigated [10]. Dataset choice remains critical: classic benchmarks such as KDD Cup’99 and NSL-KDD are historically important but may not represent real-world traffic characteristics, motivating careful selection of modern datasets and leakage-aware protocols [11].

Deep learning and hybrid learning pipelines have shown strong promise in improving detection performance by learning discriminative patterns directly from traffic features or sequences. Convolutional Neural Network (CNN)-based intrusion detection has been widely explored due to its ability to automatically extract informative patterns and reduce reliance on manual feature design [12]. Several recent studies report strong performance using deep models, feature engineering, or hybrid frameworks, while also noting practical

challenges such as class imbalance, computational cost, and generalization [13–16]. For example, CNN-based approaches have achieved high multi-class and binary detection accuracy on CIC IoT-DIAD 2024 [17], while edge-oriented anomaly detection has been explored using autoencoders with transfer learning [18]. Hybrid CNN feature extraction with XGBoost classification, combined with SHAP explanations, has also demonstrated very high accuracy for device identification and attack detection [19]. Other directions include SDN-IoT anomaly detection using DNN-integrated controllers [20], as well as tabular deep learning (TabNet) for intrusion detection and generalization analysis [21]. Additionally, efficient processing techniques (e.g., adaptive quantization) have been investigated to improve deployment practicality without compromising detection accuracy [22].

Beyond accuracy, explainability is increasingly viewed as a deployment requirement in security contexts. XAI can support analyst trust, facilitate triage, and assist with model auditing by clarifying which features contribute most to predictions [23]. Common explanation techniques include LIME and SHAP [24], and feature-importance behavior has been compared across linear and nonlinear models [25]. XAI has also been applied to improve interpretability in host-based intrusion detection and anomaly explanation, including validation/perturbation analyses and reference-based explanation strategies [26,27]. Lightweight, explanation-driven compression methods have additionally been proposed to reduce model size while preserving performance [28]. Grad-CAM-inspired explainable approaches have also been explored for intrusion detection to make deep model decisions more interpretable [1]. Recent contributions further reflect ongoing interest in IoT and cyber-physical security, spanning large-scale intrusion detection and traffic assessment, DoS detection in IoT environments [29], integrity monitoring for embedded neural networks, and UAV system security [30–33].

Table 1 summarizes representative studies using the CIC IoT-DIAD 2024 dataset and highlights the strong performance of deep and hybrid pipelines, alongside ongoing challenges such as generalization, resource efficiency, and trustworthy explanations.

Despite the reported performance of recent IDS models, data leakage is not always explicitly controlled, particularly when supervised feature selection is performed prior to train/test partitioning. If features are ranked using the full dataset, information from samples that later appear in the evaluation split can influence the selected subset and inflate performance estimates, especially under class imbalance where accuracy may remain high while minority-class performance degrades. For this reason, leakage-aware feature selection is treated in this study as a necessary step for reliable evaluation, and results are reported under both a leakage-prone (biased) protocol and a leakage-aware (unbiased) protocol to quantify the impact of selection timing on generalization, alongside macro-averaged metrics and SHAP-based model explanations of the CNN embedding used by XGBoost.

Motivated by these findings, we present an explainable hybrid IDS for CIC IoT-DIAD 2024 using 1D-CNN embeddings and XGBoost that targets strong detection performance with interpretable outputs. Our main contribution is leakage-aware evaluation rather than a novel architecture, via biased vs. unbiased feature-selection comparisons and SHAP analysis of dominant latent dimensions [23,24].

This paper is organized as follows. Section 2 presents the overall system workflow and data acquisition process, including network feature extraction, preprocessing, and feature selection. Section 3 reports the experimental results and discussion, including classification performance and explainability analysis. Finally, Section 4 concludes the paper and outlines directions for future work.

Table 1. Comparison of research studies using the CIC IoT-DIAD 2024 dataset.

Study	Task / Goal	Model(s)	Explainability	Performance	Key Strengths
[17] Hossain (2025)	Multi-class attack classification and binary anomaly detection	1D-CNN, LSTM, RNN, MLP	None	99.12% (multi-class), 99.53% (binary)	CNN gives the highest accuracy; scalable and real-time friendly.
[18] Regnell (2024)	Device-specific anomaly detection for edge IoT	Autoencoder and transfer learning (frozen encoder and fine-tuned decoder)	None	Higher AUC and F1 across 12 attacks	Lightweight; excellent for edge devices; transfer learning improves performance.
[19] Jain (2025)	IoT device identification and attack detection	CNN for feature extraction and XGBoost classifier	SHAP	99.9% device and attack classification	Hybrid model with explainability; identifies influential features.
[20] Lakshan (2025)	Anomaly detection in SDN-IoT environment	Deep neural network integrated with SDN controller	None	Over 99% detection accuracy	Real-time SDN deployment; outperforms classical ML methods.
[21] TabNet Study (2024)	Multi-class and binary intrusion detection; generalization analysis	TabNet (attentive tabular deep learning)	Feature masks (implicit interpretability)	84.6% (multi-class), 92.74% (binary); poor generalization to CIC-IDS2017	Performs well on DIAD; highlights cross-dataset generalization limitations.

2. System Overview and Data Acquisition

Figure 2 summarizes the end-to-end workflow adopted in this study. Network traffic generated by IoT devices is first captured as packet traces (PCAP) and then transformed into structured tabular records (CSV) for machine learning analysis. The workflow consists of network feature extraction, data preprocessing and feature selection, and downstream learning for intrusion detection and multi-class attack classification.

The CIC IoT-DIAD 2024 dataset [34], is used as the primary benchmark. Compared to traditional IDS benchmarks such as KDD Cup'99 and NSL-KDD, which are widely used for historical comparison, CIC IoT-DIAD 2024 is intended to reflect contemporary IoT environments. It was collected in an IoT testbed with diverse device types and a wide range of IoT-relevant attack behaviors, making it suitable for multi-class intrusion detection under realistic traffic characteristics [11,34]. In addition, it is a dual-function dataset designed for IoT device identification and anomaly/attack detection, collected in an IoT topology comprising 105 devices and including 33 distinct attacks conducted at the Canadian Institute for Cybersecurity. These attacks are grouped into seven high-level categories: DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai [34]. In the present work, emphasis is placed on intrusion detection and multi-class attack classification using the labels provided in the selected dataset partition, while device identification is treated as an optional comparison task when alignment with prior work is required.

Although the dataset includes multiple attack types, the class distribution in the selected partition is highly imbalanced. In particular, DoS/DDoS flooding categories account for most traffic records, whereas several web- and brute-force-related attacks occur with substantially fewer samples. Table 2 reports the sample count and corresponding percentage for each class in the selected partition. To further clarify the binary composition, the benign-to-malicious ratio is also reported based on the same class totals; for this partition, it is 1:18.7 (Benign:Malicious), computed from Table 2. This imbalance is important when

interpreting macro-averaged metrics, since macro-F1 assigns equal weight to each class and can therefore be disproportionately influenced by low-support minority classes.

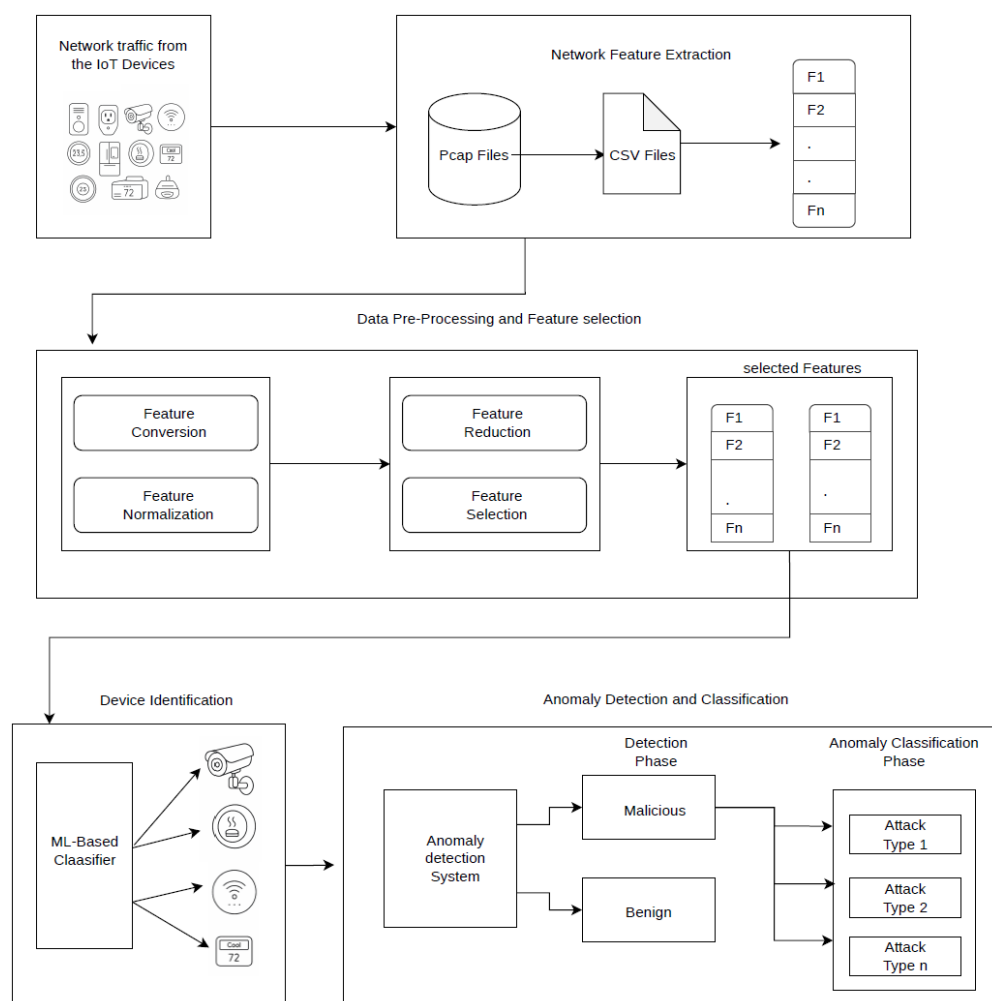


Figure 2. Overall system workflow: IoT traffic capture, network feature extraction (PCAP → CSV), preprocessing and feature selection, and downstream intrusion detection and multi-class attack classification.

Table 2. Number and percentage of samples per class in the selected CIC IoT-DIAD 2024 partition.

Class Name	Samples (N)	Percentage (%)
DOS SYN Flood	7,998,268	55.2%
DDoS ACK Fragmentation	2,449,972	16.9%
DoS HTTP Flood	1,563,223	10.8%
Benign	734,380	5%
DDoS HTTP Flood	504,597	3.4%
Vulnerability Scan	441,031	3%
DDoS ICMP Fragmentation	320,596	2.2%
DDoS ICMP Flood	196,296	1.3%
Mirai Greeth Flood	169,348	1.1%
DNS Spoofing	74,795	0.5%
SQL Injection	6600	0.045%
Dictionary Brute Force	3619	0.025%
XSS	3377	0.023%
Uploading Attack	1348	0.009%

The dataset provides both packet-based and flow-based feature representations extracted from PCAP files. Flow-based features are extracted using standard flow metering tools such as CICFlowMeter and are intended primarily for anomaly detection and attack classification, whereas packet-based features are derived from packet-level analysis and can be used for both device identification and anomaly/attack detection. The extracted features cover protocol and statistical descriptors such as flow duration, packet-length statistics, and inter-arrival timing, as well as higher-layer attributes relevant to IoT fingerprinting and security analytics such as TLS handshake fields, HTTP host and user-agent strings, DNS characteristics, and stream/jitter/channel metrics computed over multiple time intervals.

Prior to learning, the CSV tables are processed in a scalable manner suitable for large datasets. Data are read and processed in manageable blocks, then concatenated after cleaning and transformation. Preprocessing includes handling missing values, converting categorical/string fields via label encoding where applicable, and scaling numeric features using standardization to improve optimization stability and classifier performance. Labels are encoded into integer classes to support both binary (benign vs. malicious) and multi-class (attack-type) learning settings.

To improve the reliability of the evaluation, a leakage-aware protocol is applied so that feature selection and normalization are performed using training data only within each split, and the resulting transformations are then applied to the corresponding test data. In addition, a leakage-aware feature-selection strategy is considered by contrasting a biased selection that selects features using the full dataset prior to splitting against an unbiased selection that selects features using training-only information, thereby reducing label leakage and over-optimistic performance estimates. Following hybrid intrusion-detection pipelines reported on CIC IoT-DIAD, a compact feature set can be constructed by ranking features using model-based importance and retaining the top- k features for downstream learning [19]. Correlation analysis between selected features and labels can further assist in identifying overly label-correlated attributes that may indicate leakage or dataset artifacts.

After feature selection, a hybrid learning approach is employed in which a CNN learns a compact representation from the selected tabular features, and a gradient-boosted decision tree classifier XGBoost is used for final classification. Explainability is integrated using SHAP values to quantify feature contributions and analyze feature interactions for multi-class decisions, enabling both global interpretability, overall feature relevance, and local interpretability, per-sample decision rationale, in the intrusion-detection setting.

In this work, the CNN is used as a supervised representation learner that maps the selected predictors into a compact latent space to capture nonlinear feature interactions prior to downstream classification. XGBoost is then applied to the learned embedding, leveraging its strong performance on structured data and its ability to model complex decision boundaries. A direct ablation comparing XGBoost trained on the selected features (no CNN) versus XGBoost trained on CNN-derived embeddings was conducted, and the corresponding results are reported in Section 3.

3. Experimental Results and Discussion

An end-to-end pipeline was implemented to evaluate a hybrid IDS for multi-class attack classification using the CIC IoT-DIAD 2024 dataset. The original dataset comprised 84 traffic features and was refined through preprocessing to enable reliable feature extraction. Features with zero variance were removed and missing values were addressed, reducing the dimensionality to 73 features. To support efficient processing of the large dataset, chunk-wise feature scaling was adopted: the data were partitioned into manageable subsets, and the global mean and variance were incrementally updated across subsets.

The learned global mean and variance were then applied consistently to all subsets, which were subsequently recombined into a single normalized dataset for modeling.

The pipeline was implemented with scalability in mind, using block-wise ingestion during preprocessing and reducing the input dimensionality to a compact top- k subset prior to downstream learning. After the normalization parameters and selected predictors are determined within a given split, inference operates on a fixed-length feature vector per record, which supports streaming or near-real-time scoring when the same feature-extraction procedure is available. Applicability to other IoT intrusion datasets is expected when comparable flow- or packet-derived features and consistent label definitions are provided; in such cases, the same preprocessing, leakage-aware feature ranking, and CNN–XGBoost stages can be applied without structural changes, while cross-dataset evaluation is used to quantify transferability under distribution shift.

Label encoding was applied to the attack classes by mapping each categorical class name to a numerical identifier, thereby enabling supervised learning with multi-class classifiers. Table 3 reports the class names and their corresponding encoded labels. After data cleaning, an additional dimensionality-reduction stage was carried out using supervised feature selection. The objective was to retain a compact subset of predictors that preserves strong discriminative power for multi-class attack-type classification, while avoiding over-optimistic performance estimates that can arise when label-related information inadvertently leaks into the selection process. Accordingly, a top- k subset ($k = 25$) was retained based on RF-derived feature-importance ranking with respect to the encoded attack labels.

Table 3. Label encoding of the categorical attack classes.

Class Name	Encoded Label
Benign	0
DDoS ACK Fragmentation	1
DDoS HTTP Flood	2
DDoS ICMP Flood	3
DDoS ICMP Fragmentation	4
DNS Spoofing	5
DoS SYN Flood	6
Dictionary Brute Force	7
DoS HTTP Flood	8
Mirai Greeth Flood	9
SQL Injection	10
Uploading Attack	11
Vulnerability Scan	12
XSS	13

Although the evaluated dataset partition contains multiple attack types, the class distribution is highly imbalanced, with a small number of traffic categories dominating the records while several attack classes are comparatively under-represented. To make the imbalance explicit, Table 2 reports the number of samples per class.

Given the high dimensionality and heterogeneity of network traffic attributes, supervised feature selection is often adopted to reduce redundancy, improve computational efficiency, and focus learning on the most informative predictors for attack discrimination. However, feature selection is also a common source of inadvertent information leakage when the ranking step is performed using data that later appear in the evaluation split, which can bias the reported performance upward and weaken the reliability of comparisons between studies [35]. For this reason, two feature-selection protocols were evaluated; the protocols differed only in the timing of feature ranking relative to the train/test partitioning:

- (i) *Biased (leakage-prone) protocol*: Feature importance was computed using the full dataset prior to any train/test split. As a result, samples that subsequently appear in the evaluation set can influence the ranking step, which may inflate reported performance and lead to an inaccurate characterization of generalization.
- (ii) *Unbiased (leakage-aware) protocol*: Feature importance was computed using the training subset only, after partitioning. The resulting ranked list was then used to select the top- k predictors, and the same selected subset was applied to the corresponding held-out test subset. This protocol preserves evaluation-set independence and yields a more defensible estimate of performance on unseen samples.

While the above protocols explicitly address leakage arising from the timing of supervised feature selection, additional leakage mechanisms may also be present in IoT intrusion datasets. Temporal correlation may arise when temporally adjacent or near-duplicate records are distributed across training and test sets. Device overlap may occur when samples from the same device appear in both splits. Session or flow dependence may also be introduced when multiple records from the same capture context are separated across splits. Mitigation of these effects typically requires time-aware partitioning and group-aware splitting based on device or session identifiers, when such metadata are available. In this work, leakage mitigation is limited to feature-selection leakage; thus results under random K -fold CV are interpreted as estimates, not guarantees for unseen devices or time.

Feature importance was estimated using an RF model with 100 DTs, where Gini impurity was adopted as the split criterion. RF feature-importance scores were computed using impurity-based importance and aggregated across trees to produce a single ranking, after which the top- k predictors were retained. To maintain scalability on large CSV tables, importance estimation and subsequent processing were executed in a chunk-wise manner using blocks of 100,000 records. In addition, K -fold cross-validation was used to form training and testing partitions. Under the biased protocol, five folds were employed. Under the unbiased protocol, three folds were used to reduce computational overhead, since feature ranking must be repeated using training data only within each split. It should be noted that using different fold counts can introduce a potential confound in direct comparisons, since cross-validation variance and the effective training fraction differ across K . The lower- K setting in the unbiased protocol was adopted as a practical constraint due to repeated within-split feature ranking, and it is generally conservative because fewer folds can slightly reduce performance estimates. To isolate the effect of leakage-aware feature selection from the choice of fold count, we additionally report a matched-fold comparison in Section 3, where both protocols are evaluated under the same 3-fold cross-validation.

For transparency and reproducibility, the selected predictors are summarized in Table 4. The table reports the union of the top- k features obtained under both protocols and indicates which features were retained by each protocol. Several predictors were consistently selected, including flow inter-arrival statistics, throughput/rate descriptors, packet-length measures, and port identifiers, suggesting that these attributes provide stable class-discriminative information for IoT attack classification. Differences between the two ranked lists are expected, since ranking on the full dataset, the biased protocol, can implicitly incorporate evaluation-set characteristics and alter the resulting importance ordering, whereas training-only ranking, the unbiased protocol, constrains selection to patterns supported by the training distribution and therefore provides a leakage-aware basis for downstream evaluation.

Table 4. Union of top-ranked predictors selected using biased and unbiased protocols.

Feature	Category	Biased	Unbiased
ack flag count	Flags/Control	✓	
average packet size	Packet size/length	✓	✓
bwd header length	Header/structure	✓	✓
bwd iat total	Inter-arrival (bwd)	✓	
bwd packets/s	Rate/throughput	✓	✓
dst port	Ports	✓	✓
flow bytes/s	Rate/throughput	✓	✓
flow duration	Flow timing	✓	✓
flow iat max	Inter-arrival (flow)	✓	✓
flow iat mean	Inter-arrival (flow)	✓	✓
flow iat min	Inter-arrival (flow)	✓	✓
flow iat std	Inter-arrival (flow)	✓	
flow packets/s	Rate/throughput	✓	✓
fwd header length	Header/structure	✓	✓
fwd iat max	Inter-arrival (fwd)	✓	
fwd iat mean	Inter-arrival (fwd)	✓	
fwd iat min	Inter-arrival (fwd)	✓	✓
fwd iat total	Inter-arrival (fwd)	✓	
fwd init win bytes	Window/control		✓
fwd packet length max	Packet size/length		✓
fwd packet length mean	Packet size/length		✓
fwd packet length min	Packet size/length		✓
fwd packets/s	Rate/throughput	✓	✓
fwd seg size min	Header/structure		✓
fwd segment size avg	Header/structure		✓
packet length mean	Packet size/length	✓	✓
packet length min	Packet size/length	✓	✓
psh flag count	Flags/Control	✓	
rst flag count	Flags/Control		✓
src port	Ports	✓	✓
subflow bwd bytes	Flow structure	✓	
syn flag count	Flags/Control		✓
packet length max	Flow structure		✓
total length of bwd packet	Packet size/length	✓	

To further characterize the selected subsets, Pearson correlation analysis was computed for the top-*k* features obtained under each protocol with respect to the encoded attack labels. The corresponding correlation heatmaps are shown in Figure 3 and summarize linear dependencies among the selected predictors. The strongest correlation patterns occur within semantically similar feature groups, most notably among flow inter-arrival statistics and packet-length descriptors. Rate and throughput features also exhibit consistent coupling, indicating partial redundancy within the selected set while still preserving class-discriminative information for downstream learning.

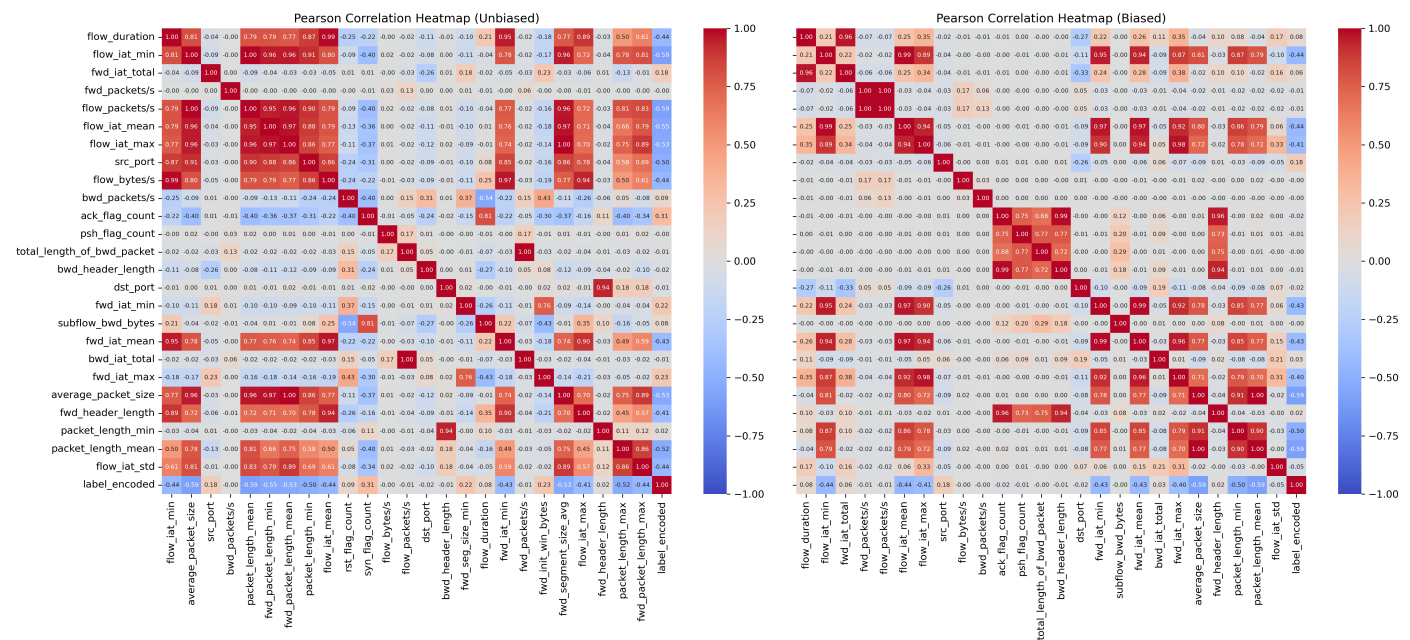


Figure 3. Pearson correlation heatmaps for the top- k features selected using the biased and unbiased feature-selection protocols.

Two CNN models with identical architectures and training settings were trained using the RF-selected feature subsets obtained under the biased and unbiased protocols. In both cases, the CNN was trained in a supervised manner and subsequently used as a deep feature extractor: the penultimate dense layer produced a 128-dimensional embedding for each traffic record, which served as the learned representation for downstream classification.

To qualitatively assess the separability of the learned embeddings, the 128-dimensional CNN outputs were projected to two dimensions using t-distributed stochastic neighbor embedding (t-SNE). The resulting visualization is shown in Figure 4. It should be noted that t-SNE is used here as a qualitative visualization tool, and the observed separation patterns are illustrative rather than definitive. When the biased (leakage-prone) feature subset was used, the projected clusters appeared to exhibit clearer separation and reduced overlap among several classes. This visual difference is reported only as a qualitative observation and is not used as definitive evidence of leakage. Because the biased protocol computes feature ranking using the full dataset prior to evaluation, it can incorporate test-set information into the selected feature subset; therefore, any conclusions regarding leakage are grounded in the experimental protocol definition and the quantitative cross-validation results, rather than the t-SNE visualization. In contrast, the embeddings obtained with the unbiased (leakage-aware) feature subset showed less pronounced separation and more controlled overlap across classes, which is consistent with a more conservative representation learned under a strictly held-out feature-ranking protocol.

The CNN architecture and hyperparameters used for feature extraction are summarized in Table 5. Training was performed for 30 epochs using the Adam optimizer. During training, the CNN includes the final softmax layer (Dense-2) and is optimized end-to-end for 30 epochs using a softmax cross-entropy loss with the Adam optimizer. After training, the softmax layer is removed and the output of the penultimate Dense-1 layer is used as a 128-dimensional embedding for feature extraction.

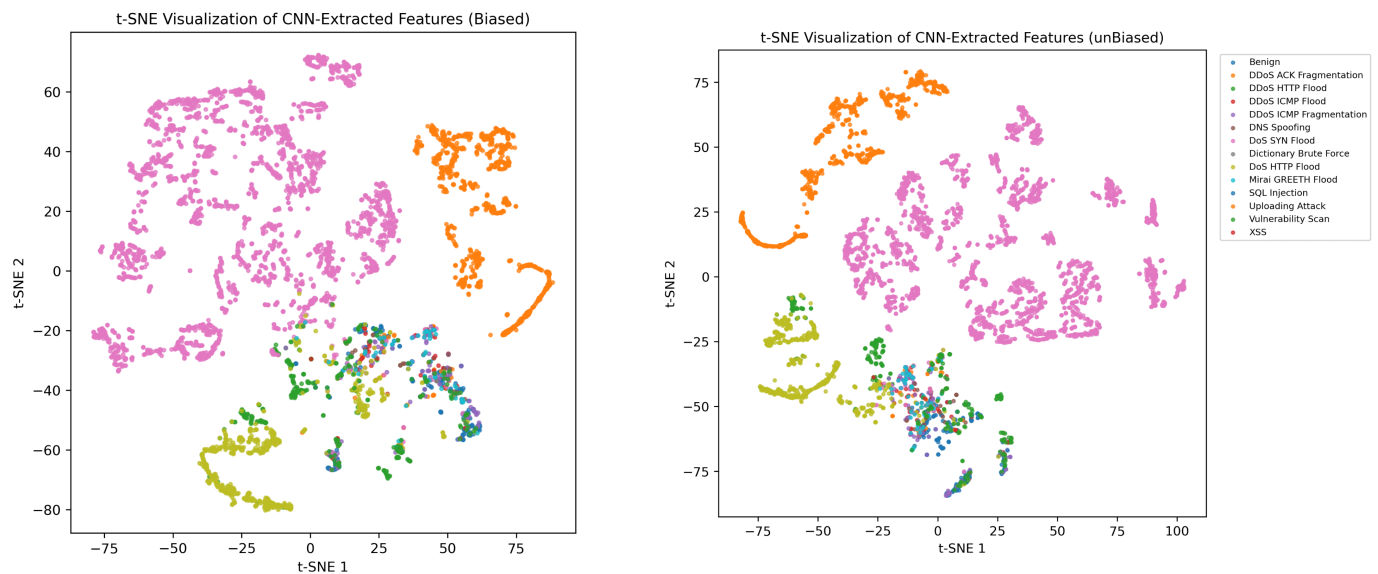


Figure 4. t-SNE visualization of the CNN-extracted embeddings under the biased and unbiased feature-selection protocols.

Table 5. CNN model parameters used for feature extraction.

Layer	Type	Parameters	Output Dimension
Input	Input layer	25 features	(25, 1)
Conv1D-1	Convolutional	64 filters, kernel size = 3, ReLU	(23, 64)
MaxPool-1	Max pooling	pool size = 2	(11, 64)
Conv1D-2	Convolutional	128 filters, kernel size = 3, ReLU	(9, 128)
MaxPool-2	Max pooling	pool size = 2	(4, 128)
Flatten	Flatten	—	512
Dense-1	Fully connected	128 neurons, ReLU	128
Dropout	Regularization	rate = 0.3	128
Dense-2	Fully connected	Softmax	C = 14

A 1D-CNN was selected because the input to the network is a one-dimensional vector of engineered tabular predictors; convolution along this feature axis provides a lightweight mechanism to learn local feature interactions and nonlinear combinations without imposing an artificial 2D/3D spatial structure that is more appropriate for images or volumetric data.

The selected predictors are tabular and therefore do not possess an inherent spatial topology as in images; accordingly, no claim is made that the feature axis encodes a physical neighborhood structure. Instead, the 25-dimensional input is provided to the network in a fixed, deterministic order (kept identical across training and evaluation), and 1D convolutions are employed primarily as a parameter-sharing mechanism to learn nonlinear feature interactions with fewer parameters than a comparably sized fully connected MLP. To support this design choice and to quantify the contribution of the representation-learning stage, an ablation study was conducted under the same evaluation setting by comparing against two baselines trained directly on the selected features: (i) a standard MLP and (ii) a direct XGBoost model without the CNN embedding stage. Table 6 summarizes the aggregate results, indicating that the CNN embeddings → XGBoost pipeline yields higher macro-F1 than these baselines while maintaining comparable accuracy and weighted-F1.

Table 6. Ablation on the selected tabular predictors: comparison of baselines trained directly on the selected features (MLP and XGBoost without CNN) versus the proposed CNN embeddings → XGBoost pipeline (aggregate metrics).

Variant	Accuracy	Macro F1-Score	Weighted F1-Score
XGBoost on selected features (no CNN)	0.92	0.52	0.92
MLP on selected features	0.91	0.43	0.91
Proposed: CNN embeddings → XGBoost	0.93	0.59	0.93

The 128-dimensional embedding produced by the CNN feature extractor was used as the input representation for the final classifier. Classification was performed using XGBoost, a gradient-boosted decision tree ensemble in which trees are added sequentially and each new tree is fitted to reduce the residual errors of the current ensemble. XGBoost was used with its standard implementation settings and a multi-class objective (softprob), with the number of classes set to $C = 14$. In this setting, the CNN acts as a representation learner that maps the selected tabular features into a compact latent space, while XGBoost operates on this latent space to learn nonlinear decision boundaries between the attack classes. Consistent with the two feature-selection protocols, two XGBoost classifiers were trained: one using embeddings extracted from the CNN trained on the biased feature subset, and one using embeddings extracted from the CNN trained on the unbiased feature subset.

Model performance was quantified using accuracy as well as macro- and weighted-averaged precision, recall, and F1-score. Macro-averaged metrics assign equal weight to each class and are therefore informative under class imbalance, whereas weighted metrics reflect aggregate performance while accounting for the class support. Discriminative ability was further assessed using one-vs.-rest ROC curves, reporting the macro-averaged ROC–AUC, and using the macro-averaged precision–recall (PR) score.

Under the biased feature-selection protocol, the CNN → XGBoost pipeline achieved an accuracy of 0.9302, a macro-F1 of 0.5862, and a weighted-F1 of 0.9295. The macro-precision and macro-recall were 0.7323 and 0.5443, respectively, while the weighted precision and weighted recall were 0.9355 and 0.9302. A macro ROC–AUC of 0.9897 and a macro PR score of 0.6134 were obtained.

Under the unbiased feature-selection protocol, performance increased slightly and more consistently across the macro-averaged criteria. An accuracy of 0.9324 was achieved, with a macro-F1 of 0.5911 and a weighted-F1 of 0.9321. The macro-precision and macro-recall were 0.7422 and 0.5501, respectively, while the weighted precision and weighted recall were 0.9378 and 0.9324. The macro ROC–AUC increased to 0.9905 and the macro PR score to 0.6218.

Because the initial evaluation used different fold counts across protocols for computational tractability, a matched 3-fold comparison was additionally conducted to remove this potential confound. In particular, the biased (leakage-prone) protocol was re-evaluated under the same 3-fold cross-validation setting adopted for the unbiased (leakage-aware) protocol. Table 7 reports mean $\pm$ standard deviation across folds for key aggregate metrics, together with the corresponding deltas (Unbiased – Biased) and paired-test p -values. Under the matched 3-fold setting, the aggregate metrics are very close across protocols and the fold-wise differences are not statistically significant at $\alpha = 0.05$, indicating that the overall conclusions are not driven by the earlier fold-count mismatch.

Table 7. Matched 3-fold cross-validation comparison between biased (leakage-prone) and unbiased (leakage-aware) feature-selection protocols (mean $\pm$ standard deviation across folds). The last column reports the p -value from a two-sided paired test across folds.

Metric	Biased	Unbiased	Δ (Unbiased—Biased)	p -Value (Paired Test)
Accuracy	0.9328 ± 0.00013	0.9327 ± 0.00015	-0.0001	0.1835
Macro F1-score	0.5875 ± 0.00310	0.5898 ± 0.00213	$+0.0023$	0.0913
Weighted F1-score	0.9325 ± 0.00011	0.9324 ± 0.00024	-0.0001	0.4226

Confusion matrices for both protocols are reported in Figure 5. The dominant mass remains on the diagonal for high-support categories, while most residual errors are concentrated in a small subset of minority classes. Consistent with the per-class behavior reflected by the macro-averaged metrics, the highest misclassification rates occur for the rare web- and brute-force-related categories, including Dictionary Brute Force, XSS, Uploading Attack, and SQL Injection, whereas major DoS/DDoS classes are detected more reliably. A higher number of correct predictions was observed under the unbiased protocol, which is consistent with the expected benefit of leakage-aware feature selection: by ensuring that the test portion remains unseen during feature ranking, the resulting evaluation provides a more defensible estimate of generalization performance.

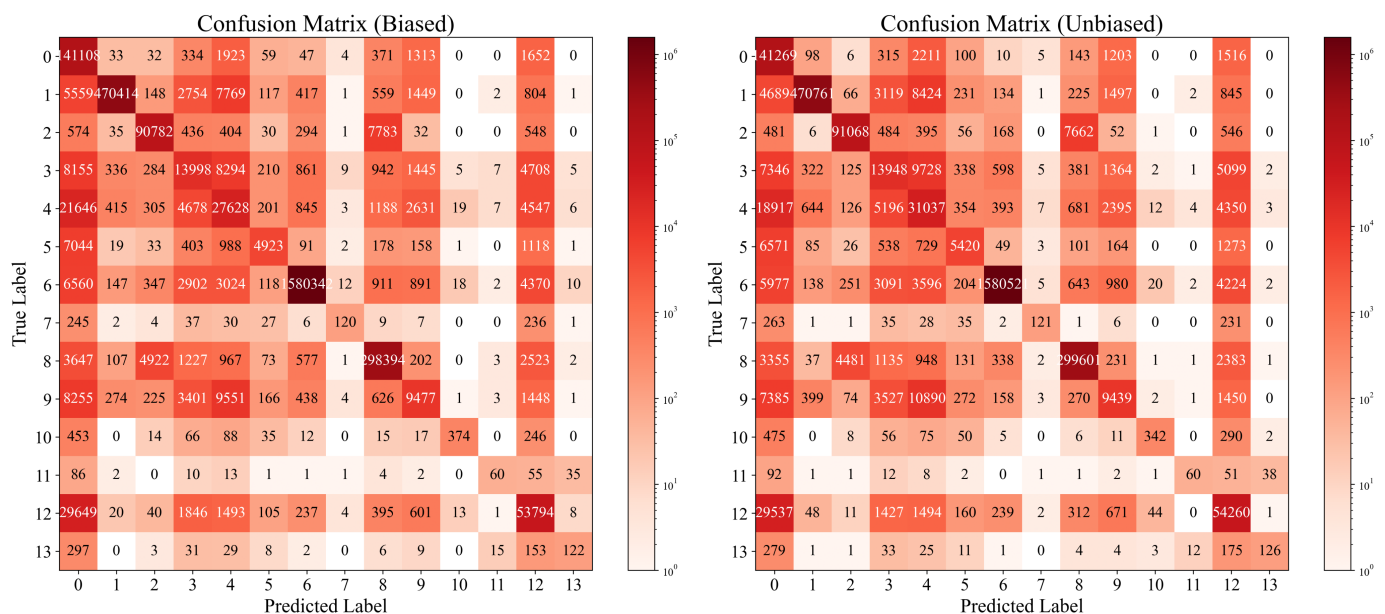


Figure 5. Confusion matrices obtained under the biased and unbiased feature-selection protocols.

Figure 6 presents the one-vs.-rest ROC curves for all attack classes together with the macro-averaged ROC-AUC for the CNN-XGBoost classifier under the biased and unbiased feature-selection protocols. In both cases, the majority of ROC traces remain close to the upper-left region of the ROC space, indicating that high true-positive rates are achieved at relatively low false-positive rates across a wide range of decision thresholds. This behavior suggests that the learned decision function provides discriminative capability beyond chance-level prediction, particularly for the high-support classes. However, the lower macro-F1 indicates that this discriminative capability is not uniform across all classes under severe imbalance; most residual errors occur in a small subset of minority web/brute-force categories (see Figure 5 and Table 8). A modest increase in macro-averaged ROC-AUC is observed for the unbiased protocol relative to the biased protocol, which is consistent with

the expectation that leakage-aware feature selection yields a more defensible estimate of performance on unseen data and reduces the risk of overly optimistic evaluation.

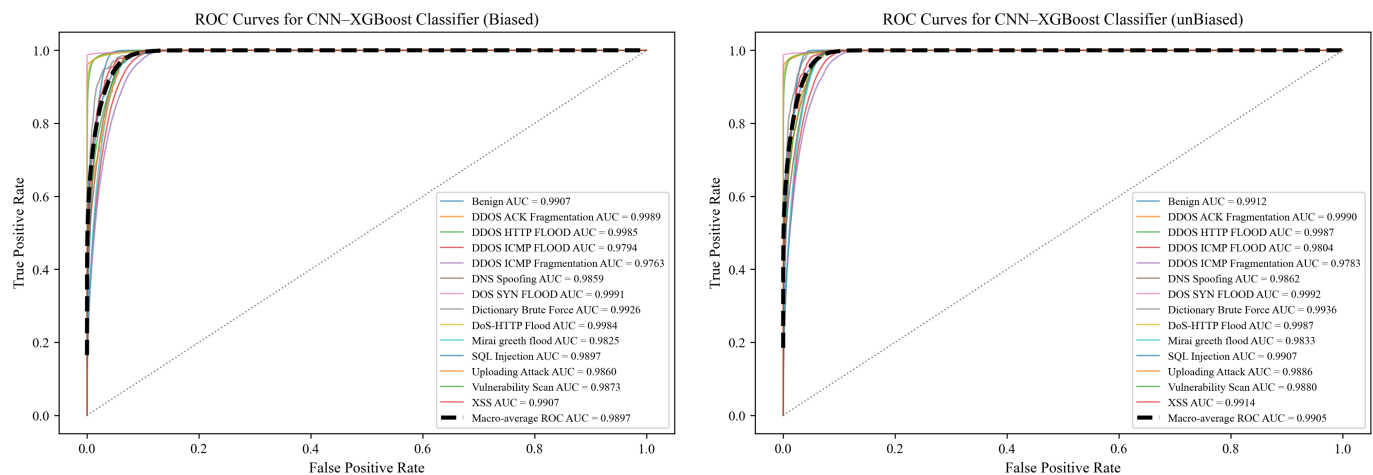


Figure 6. One-vs.-rest ROC curves and macro-averaged ROC–AUC for the CNN–XGBoost classifier under biased and unbiased feature-selection protocols.

In addition to ROC–AUC, aggregate performance was summarized using both macro-averaged and weighted-averaged metrics. A noticeable gap between these two averaging schemes was observed for both protocols. This discrepancy is commonly associated with pronounced class imbalance: weighted averages are dominated by majority classes and can therefore remain high even when minority classes exhibit lower recall and F1-scores, whereas macro averages assign equal weight to each class and are thus more sensitive to minority-class performance degradation. Consequently, macro-averaged metrics provide a more conservative view of performance in imbalanced multi-class intrusion-detection settings, while weighted metrics reflect overall effectiveness under the empirical class distribution.

Overall, the biased and unbiased protocols yielded very similar performance, with only small differences in point estimates. Under the matched 3-fold setting (Table 7), paired testing on the fold-wise deltas (Unbiased–Biased) did not indicate statistical significance at $\alpha = 0.05$; therefore, these deltas should be interpreted cautiously and may fall within cross-validation variability (noting that $K = 3$ provides limited statistical power). Rather than claiming large performance gains, we emphasize that the unbiased (leakage-aware) protocol is methodologically preferable because it enforces evaluation-set independence during feature ranking and therefore provides a more defensible generalization estimate. To provide a class-granular assessment, Table 8 reports per-class precision, recall, and F1-score for both protocols, highlighting attack categories that remain challenging and indicating where differences, if any, are most apparent.

To support interpretability, SHAP analysis was conducted to quantify feature contributions to the model decisions. Since the unbiased protocol provides the more reliable generalization estimate and achieved marginally improved overall performance, SHAP explanations were computed for the corresponding unbiased CNN–XGBoost model. This analysis enables both global interpretation (ranking the most influential predictors across the dataset) and local interpretation (explaining individual predictions and misclassifications), thereby improving transparency and supporting analyst-driven validation of the learned intrusion-detection behavior.

Table 8. Class-wise precision, recall, and F1-score comparison for unbiased and biased feature-selection protocols.

Class	Unbiased			Biased		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Benign	0.6233	0.9618	0.7564	0.6049	0.9607	0.7424
DDoS ACK Fragmentation	0.9962	0.9607	0.9782	0.9971	0.9600	0.9782
DDoS HTTP Flood	0.9462	0.9024	0.9238	0.9346	0.8996	0.9167
DDoS ICMP Flood	0.4237	0.3553	0.3865	0.4358	0.3566	0.3922
DDoS ICMP Fragmentation	0.4460	0.4841	0.4643	0.4442	0.4309	0.4374
DNS Spoofing	0.7360	0.3623	0.4856	0.8106	0.3291	0.4681
DoS SYN Flood	0.9987	0.9880	0.9933	0.9976	0.9879	0.9927
Dictionary Brute Force	0.7806	0.1671	0.2753	0.7407	0.1657	0.2709
DoS-HTTP Flood	0.9664	0.9583	0.9623	0.9583	0.9544	0.9564
Mirai Greeth Flood	0.5238	0.2787	0.3638	0.5197	0.2798	0.3638
SQL Injection	0.7991	0.2591	0.3913	0.8677	0.2833	0.4272
Uploading Attack	0.7229	0.2222	0.3399	0.6000	0.2222	0.3243
Vulnerability Scan	0.7075	0.6152	0.6581	0.7059	0.6099	0.6544
XSS	0.7200	0.1867	0.2965	0.6354	0.1807	0.2814

SHAP was used to interpret the CNN–XGBoost model trained under the unbiased (leakage-aware) protocol. Because the CNN produces a 128-dimensional latent representation, the SHAP analysis in this subsection reflects the contribution of the CNN-extracted latent features (feature indices 1–128) to the final XGBoost decision function, rather than the original network traffic variables.

Accordingly, the SHAP explanations in this subsection should be interpreted as latent-space attributions that quantify how the downstream XGBoost classifier leverages the CNN embedding dimensions to form its decisions. Although the latent dimensions are learned from the original traffic-feature vector, they represent compressed combinations of multiple inputs and therefore do not admit a one-to-one correspondence with individual traffic variables. Any security or behavioral interpretation should thus be treated as indirect unless an explicit latent-to-input attribution analysis is performed.

A practical linkage back to input variables can be obtained by treating the trained CNN as a deterministic mapping from the selected traffic features to the latent embedding. After the most influential latent dimensions are identified (e.g., by SHAP), input-level relevance can be estimated through a complementary attribution step on the CNN, such as gradient-based methods (saliency or integrated gradients) computed with respect to those dominant latent dimensions. Alternatively, controlled perturbations can be applied in the original input space while tracking the induced changes in (i) the dominant latent dimensions and (ii) the final XGBoost outputs. This provides an analyst-oriented pathway to relate influential latent factors to concrete traffic-feature indicators without altering the evaluation protocol used in this study. A quantitative implementation of these input-level mapping procedures is outside the scope of the present study and is left for future work.

Figure 7 summarizes the global distribution of absolute SHAP coefficients across all 128 latent features and all evaluated samples. A strongly right-skewed (heavy-tailed) distribution is observed: most latent dimensions exhibit near-zero contribution for the majority of samples, whereas a small subset attains substantially larger SHAP magnitudes. This concentration of attribution mass indicates that the downstream classifier relies primarily on a limited number of highly informative latent directions, while many remaining dimensions contribute marginally. Such behavior is consistent with a representation-learning stage that

compresses discriminative structure into a sparse set of salient latent factors, which may also suggest that further latent-space pruning or dimensionality sensitivity analysis could be feasible without severely degrading performance.

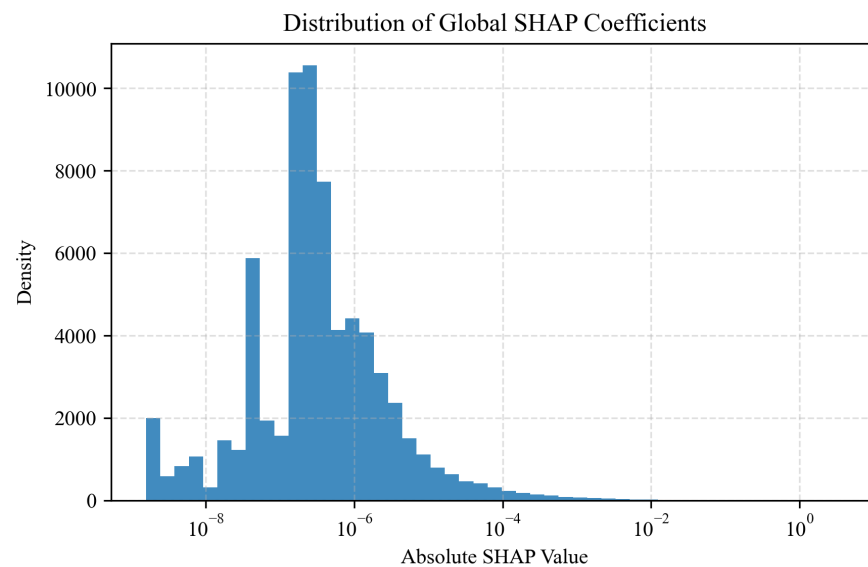


Figure 7. Global distribution of absolute SHAP coefficients for the unbiased model (latent CNN features).

To resolve how the learned representation supports specific attack categories, class-wise mean absolute SHAP values were computed and aggregated for the top 20 latent features, as shown in Figure 8. Each bar reports the mean contribution of a latent feature, with the stacked segments indicating how that feature contributes across different classes. Several latent dimensions exhibit class-dependent importance (i.e., stronger attribution for a subset of attacks), suggesting that parts of the learned representation encode patterns that are more distinctive for particular behaviors. At the same time, a consistently dominant latent dimension is apparent: feature 66 attains the highest mean impact across most classes, indicating that it captures broadly discriminative structure shared by multiple attack types and benign traffic. This global relevance suggests that feature 66 may encode a high-level factor correlated with general traffic intensity or temporal/structural irregularities that manifest across many attacks. In contrast, the remaining top-ranked latent features show comparatively more selective contributions, which is consistent with additional latent factors specializing in finer-grained class separation. Collectively, these observations indicate that the CNN representation contains both shared (cross-class) and class-specific latent cues, which are subsequently leveraged by XGBoost to form decision boundaries. From an analyst perspective, the concentration of attribution in a small set of latent dimensions also suggests that model behavior is driven by a limited number of dominant factors, which can simplify downstream auditing and robustness checks.

Overall, the SHAP results support two complementary conclusions: (i) the learned 128-dimensional CNN representation is effectively utilized in a sparse manner by the XGBoost classifier, with decision-making dominated by a small subset of latent dimensions; and (ii) within that subset, both globally discriminative latent factors (e.g., feature 66) and class-selective latent factors are present, providing evidence that the hybrid pipeline combines shared attack-related signatures with more specialized class-specific cues.

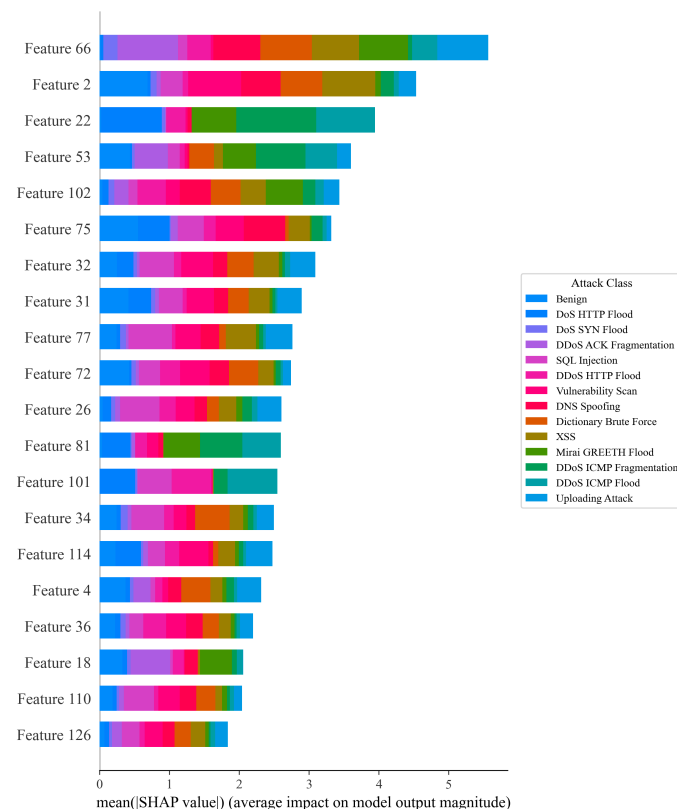


Figure 8. Class-wise mean absolute SHAP values for the top 20 latent CNN features in the unbiased model.

4. Conclusions and Future Work

This study presented an explainable CNN–XGBoost IDS pipeline for multi-class IoT attack classification on CIC IoT-DIAD 2024. The end-to-end workflow combined scalable, chunk-wise preprocessing and standardization with supervised feature reduction, retaining a compact top- k subset ($k = 25$) selected via RF importance. To explicitly address selection bias, two feature-selection protocols were compared: a biased protocol that ranks features on the full dataset prior to splitting, and an unbiased protocol that performs ranking using training data only within each split. The results showed that the unbiased protocol yields a more defensible evaluation because it enforces evaluation-set independence during feature ranking; however, under the matched 3-fold setting, the protocol-level deltas were small and did not reach statistical significance at $\alpha = 0.05$ (paired test with limited power), and should therefore be interpreted cautiously within cross-validation variability. Class-wise analysis further highlighted the impact of class imbalance, where weighted metrics remained high while macro metrics exposed persistent challenges for several minority attack classes. In practical IDS workflows, the proposed protocol-level comparison provides a simple diagnostic to quantify how feature-ranking timing can affect reported performance, supporting more reliable evaluation prior to deployment.

In addition, an ablation on tabular backbones (MLP and direct XGBoost on the selected features) was conducted, and the results are reported in Section 3, supporting the contribution of the CNN embeddings $\rightarrow$ XGBoost representation stage on this dataset.

Model transparency was incorporated through SHAP explanations computed for the unbiased model. Because the final classifier operates on 128-dimensional CNN embeddings, the SHAP analysis quantified the contribution of latent CNN dimensions to the XGBoost decision function rather than the original traffic variables. The attributions exhibited a heavy-tailed distribution, indicating that model decisions are dominated by a small subset

of influential latent factors. Class-wise aggregation further suggested that these factors include both globally discriminative latent directions shared across multiple attacks and class-selective latent cues that support finer-grained separation. Overall, the findings indicate that leakage-aware feature selection combined with CNN-based representation learning, tree-based classification, and latent-space explainability can provide competitive multi-class IDS performance together with interpretable decision support. At the same time, interpretability is currently provided at the latent level, and under the matched 3-fold setting the protocol-level deltas were not statistically significant at $\alpha = 0.05$ (paired test with limited power), so they should be interpreted cautiously within cross-validation variability.

Future work will strengthen the interpretability and external validity of the proposed approach. First, latent attributions will be connected back to the original traffic variables by reporting input-feature-level explanations (e.g., SHAP on the selected feature space and complementary analyses that relate dominant latent dimensions to salient inputs) and by presenting clearer class-name mappings where needed. Second, robustness will be assessed under more leakage-resistant evaluation settings, including device-wise and time-wise splits, alongside sensitivity studies over k and additional ablations isolating the contribution of feature selection, the CNN representation stage, the XGBoost classifier, and the explainability analysis. Finally, cross-dataset evaluation and deployment-oriented profiling will be considered to quantify transferability, computational cost, and suitability for resource-constrained IoT/edge environments. In addition, latency and throughput profiling under streaming ingestion will be reported to better characterize scalability for real-time detection scenarios.

Author Contributions: Conceptualization, D.A., A.S.K. and L.M.; methodology, D.A., A.S.K. and L.M.; software, D.A.; validation, D.A.; formal analysis, D.A.; investigation, D.A.; resources, A.S.K.; data curation, D.A., A.S.K. and L.M.; writing—original draft preparation, D.A.; writing—review and editing, A.S.K. and L.M.; visualization, D.A.; supervision, A.S.K. and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset analyzed in this study is publicly available from the Canadian Institute for Cybersecurity (CIC). Derived data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors would like to thank Gandeve Satria, at the College of Computer and Systems Engineering, Abdullah Al-Salem University, for their valuable input.

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

References

1. Sun, Y.; Lian, Z.; Zhang, S.; Wang, Z.; Duan, T. A Novel Explainable Method based on Grad-CAM for Network Intrusion Detection. In *Proceedings of the 2024 IEEE 24th International Conference on Software Quality, Reliability and Security (QRS)*; IEEE: New York, NY, USA, 2024; pp. 400–406. [[CrossRef](#)]
2. Ferrario, A.; Loi, M. How Explainability Contributes to Trust in AI. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*, Seoul, Republic of Korea, 21–24 June 2022; pp. 1457–1466. [[CrossRef](#)]
3. Ashoor, A.S.; Gore, S. Importance of Intrusion Detection System (IDS). *Int. J. Sci. Eng. Res.* **2011**, *2*, 73–76.
4. Potteti, S.; Parati, N. Intrusion Detection System Using Hybrid Fuzzy Genetic Algorithm. In *Proceedings of the 2017 International Conference on Trends in Electronics and Informatics (ICEI)*, Tirunelveli, India, 11–12 May 2017; pp. 613–618. [[CrossRef](#)]
5. Li, Y.; Liu, Q. A Comprehensive Review Study of Cyber-Attacks and Cyber Security: Emerging Trends and Recent Developments. *Energy Rep.* **2021**, *7*, 8176–8186. [[CrossRef](#)]

6. Gangwar, A.; Sahu, S. A Survey on Anomaly and Signature Based Intrusion Detection System. *Int. J. Eng. Res. Appl. (IJERA)* **2014**, *4*, 67–72.
7. Alias, S.B.; Manickam, S.; Kadhun, M.M. A Study on Packet Capture Mechanisms in Real Time Network Traffic. In Proceedings of the 2013 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuching, Malaysia, 23–24 December 2013; pp. 456–460. [\[CrossRef\]](#)
8. Isa, M.M.; Mhamdi, L. Native SDN Intrusion Detection using Machine Learning. In *Proceedings of the 2020 IEEE Eighth International Conference on Communications and Networking (ComNet)*; IEEE: New York, NY, USA, 2020; pp. 1–7. [\[CrossRef\]](#)
9. Isa, M.M.; Mhamdi, L. Hybrid Deep Autoencoder with Random Forest in Native SDN Intrusion Detection Environment. In *Proceedings of the ICC 2022—IEEE International Conference on Communications*; IEEE: New York, NY, USA, 2022; pp. 1698–1703. [\[CrossRef\]](#)
10. Babatope, L.O.; Babatunde, L.; Ayobami, I. Strategic Sensor Placement for Intrusion Detection in Network-Based IDS. *Int. J. Intell. Syst. Appl.* **2014**, *6*, 61–68. [\[CrossRef\]](#)
11. Mohd Yusof, N.N.; Sulaiman, N.S. Cyber Attack Detection Dataset: A Review. *J. Phys. Conf. Ser.* **2022**, *2319*, 012029. [\[CrossRef\]](#)
12. Ho, S.; Al Jufout, S.; Dajani, K.; Mozumdar, M. A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks Using Convolutional Neural Network. *IEEE Open J. Comput. Soc.* **2021**, *2*, 14–25. [\[CrossRef\]](#)
13. Gaber, M.G.; Ahmed, M.; Janicke, H. Malware Detection with Artificial Intelligence: A Systematic Literature Review. *ACM Comput. Surv.* **2024**, *56*, 1–33. [\[CrossRef\]](#)
14. Ali, A.; Zia, A.; Razzaque, A.; Shahid, H.; Sheikh, H.T.; Saleem, M.; Yousaf, F.; Muneer, S. Enhancing Cybersecurity with Artificial Neural Networks: A Study on Threat Detection and Mitigation Strategies. In *Proceedings of the 2024 2nd International Conference on Cyber Resilience (ICCR)*; IEEE: New York, NY, USA, 2024; pp. 1–5. [\[CrossRef\]](#)
15. Jang, M.; Lee, K. An Advanced Approach for Detecting Behavior-Based Intranet Attacks by Machine Learning. *IEEE Access* **2024**, *12*, 52480–52495. [\[CrossRef\]](#)
16. Kohli, M.; Chhabra, I. A comprehensive survey on techniques, challenges, evaluation metrics and applications of deep learning models for anomaly detection. *Discov. Appl. Sci.* **2025**, *7*, 784. [\[CrossRef\]](#)
17. Hossain, M.A. Deep Learning-Based Intrusion Detection for IoT Networks: A Scalable and Efficient Approach. *Eurasip J. Inf. Secur.* **2025**, *2025*, 28. [\[CrossRef\]](#)
18. Regnell, R. Anomaly Detection on Edge Networks. Master's Thesis, Lund University, Lund, Sweden, 2025.
19. Jain, P.; Rathour, A.; Sharma, A.; Chhabra, G.S. Bridging Explainability and Security: An XAI-Enhanced Hybrid Deep Learning Framework for IoT Device Identification and Attack Detection. *IEEE Access* **2025**, *13*, 127368–127390. [\[CrossRef\]](#)
20. Lakshan Yasarathna, T.; Liyanage, M.; Le-Khac, N.A. Deep Learning-Based Autonomous Anomaly Detection for Security in SDN-IoT Networks. *IEEE Open J. Commun. Soc.* **2025**, *6*, 8007–8048. [\[CrossRef\]](#)
21. Lazuardi, A.F.; Wibowo, S.A.; Karna, N. Generalization Performance of Internet of Things Intrusion Detection System Built on Impact-based Dataset Using TabNet Architecture. In *Proceedings of the 2025 International Conference on Data Science and Its Applications (ICoDSA)*; IEEE: New York, NY, USA, 2025; pp. 1309–1314.
22. Baldini, G.; Sithamparanathan, K. Intrusion Detection System for MIL-STD-1553 Based on Convolutional Neural Networks With Binary Images and Adaptive Quantization. *IEEE Netw. Lett.* **2024**, *6*, 50–54. [\[CrossRef\]](#)
23. Rjoub, G.; Bentahar, J.; Abdel Wahab, O.; Mizouni, R.; Song, A.; Cohen, R.; Otrouk, H.; Mourad, A. A Survey on Explainable Artificial Intelligence for Cybersecurity. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 5115–5140. [\[CrossRef\]](#)
24. Zhang, X.; Gao, J. Measuring Feature Importance of Convolutional Neural Networks. *IEEE Access* **2020**, *8*, 196062–196074. [\[CrossRef\]](#)
25. Saarela, M.; Jauhiainen, S. Comparison of feature importance measures as explanations for classification models. *SN Appl. Sci.* **2021**, *3*, 272. [\[CrossRef\]](#)
26. Gaspar, D.; Silva, P.; Silva, C. Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron. *IEEE Access* **2024**, *12*, 30164–30175. [\[CrossRef\]](#)
27. Lee, H.W.; Han, T.H.; Lee, T.J. Reference-Based AI Decision Support for Cybersecurity. *IEEE Access* **2023**, *11*, 143324–143339. [\[CrossRef\]](#)
28. Baradaran Raji, S.; Ramezani, M.; Mousavi, S.H.; Amini, M.; Gheybi, A. An Explainability-Driven Approach for Compressing Convolutional Neural Networks. In Proceedings of the 1st National Conference on Artificial Intelligence and Software Engineering, Shiraz, Iran, 1 November 2023.
29. Jiao, Q.; Mhamdi, L. Deep Learning based Intrusion Detection for IoT Networks. In *Proceedings of the 2024 Global Information Infrastructure and Networking Symposium (GIIS)*; IEEE: New York, NY, USA, 2024. [\[CrossRef\]](#)
30. 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. [\[CrossRef\]](#)
31. Ruuhwan.; Munadi, R.; Nuha, H.H.; Setiawan, E.B.; Cahyani, N.D.W. A Statistical Method and Deep Learning Models for Detecting Denial of Service Attacks in the Internet of Things (IoT) Environment. *Appl. Syst. Innov.* **2026**, *9*, 9. [\[CrossRef\]](#)

32. Pereira, A.; Paulin, D.; Hennebert, C. Beyond Histotrust: A Blockchain-Based Alert in Case of Tampering with an Embedded Neural Network in a Multi-Agent Context. *Appl. Syst. Innov.* **2026**, *9*, 19. [[CrossRef](#)]
33. Ale Isaac, M.S.; Flores Pe na, P.; Gifu, D.; Ragab, A.R. Advanced Control Strategies for Securing UAV Systems: A Cyber-Physical Approach. *Appl. Syst. Innov.* **2024**, *7*, 83. [[CrossRef](#)]
34. Rabbani, M.; Gui, J.; Nejati, F.; Zhou, Z.; Kaniyamattam, A.; Mirani, M.; Piya, G.; Opushnyev, I.; Lu, R.; Ghorbani, A.A. Device Identification and Anomaly Detection in IoT Environments. *IEEE Internet Things J.* 2024, *early access*. [[CrossRef](#)]
35. Cawley, G.C.; Talbot, N.L.C. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.

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.