

DETECX: Deep Ensemble Temporal Encoder with Cross-Attention Mechanism for Encrypted Traffic Intrusion Detection in Cloud-Edge Collaborative Architectures

Xianjun Yang

yangxianjunhpu@outlook.com

Hawaii Pacific University

LiPeng Wang

Xinjiang Arts Institute

Research Article

Keywords:

Posted Date: October 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11252570/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

DETECX: Deep Ensemble Temporal Encoder with Cross-Attention Mechanism for Encrypted Traffic Intrusion Detection in Cloud-Edge Collaborative Architectures

Abstract

The rapid proliferation of TLS-encrypted network traffic has rendered deep packet inspection-based intrusion detection systems (IDS) fundamentally ineffective, as payload content is no longer accessible for analysis. Simultaneously, the migration of security-sensitive workloads to cloud-edge collaborative architectures introduces stringent real-time constraints that preclude the centralised deployment of computationally heavy detection models. To address these dual challenges, this paper proposes DETECX (Deep Ensemble Temporal Encoder with Cross-Attention Mechanism), a novel framework for encrypted traffic intrusion detection in cloud-edge collaborative environments. DETECX integrates three architectural innovations: a Multi-Scale Temporal Convolutional Encoder (MTCE) that simultaneously models encrypted flow-level behavioural patterns at four temporal granularities via dilated causal convolutions ($d \in \{1, 2, 4, 8\}$); a Cross-Attention Fusion Module (CAFM) that dynamically fuses cloud-side global attack semantics with edge-side local flow representations through a learned cross-domain attention mechanism; and an Ensemble Decision Module (EDM) employing three diversity-promoting subnetworks with adaptive weighted voting to improve detection recall for low-frequency attack classes. Comprehensive experiments on three publicly available benchmark datasets — CICIDS2017, CICIoT2023, and UNSW-NB15 — demonstrate that DETECX achieves an average accuracy of 99.18%, macro F1-score of 97.38%, and AUC of 99.56%, consistently outperforming seven representative baseline methods. The cloud-edge deployment achieves an edge-side inference latency of 5.8 ms per flow with a $300\times$ bandwidth reduction. Ablation studies confirm F1 improvements of 2.53, 2.03, 1.37, and 1.70 pp from MTCE, CAFM, EDM, and Cloud Layer respectively. Cross-dataset generalisation yields an average cross-domain F1 of 0.8794, with all improvements statistically significant ($p < 0.05$, Wilcoxon signed-rank test).

Keywords: Encrypted traffic intrusion detection; Deep ensemble learning; Temporal convolutional network; Cross-attention mechanism; Cloud-edge collaborative architecture; Network security

1 Introduction

Network security is undergoing a fundamental transformation driven by two simultaneous and mutually reinforcing trends, each of which independently challenges the effectiveness of conventional intrusion detection systems (IDS), and whose combination creates a security gap that existing approaches are architecturally incapable of bridging.

The first trend is the near-universal adoption of transport-layer encryption. Studies indicate that TLS-encrypted connections now constitute over 95% of web traffic in major production networks [27, 28], a proportion driven by the mandatory enforcement of TLS 1.3 in modern browsers, cloud services, and enterprise environments. While encryption is essential for protecting user privacy and data integrity, it renders the most informative traffic features — packet payloads, application-layer headers, and content signatures — permanently opaque to intrusion detection systems [32]. Deep Packet Inspection (DPI), which has underpinned network security monitoring for decades, becomes fundamentally inoperable when payloads are encrypted. Signature-based IDS [15] and content-aware anomaly detection [30] cannot inspect what they cannot see; they are reduced to operating on flow-level metadata such as packet length distributions, inter-arrival times, TCP flag counts, and flow duration statistics that provide only an indirect behavioural signature of malicious activity.

The second trend is the rapid migration of security-sensitive workloads toward cloud-edge collaborative architectures. The proliferation of Internet of Things devices, mobile edge computing deployments, and 5G network infrastructure has created vast distributed environments where traffic must be monitored in real time at billions of endpoints [25, 29]. Centralising all intrusion detection at the cloud introduces unacceptable round-trip latency and unbounded communication overhead; deploying a full detection model at each constrained edge node is computationally infeasible without careful architectural design. The vision of edge computing articulated by Shi et al. [29] and the edge intelligence paradigm proposed by Zhou et al. [35] offer a theoretical foundation, yet realising their potential for IDS requires novel architectural solutions that neither paradigm alone provides.

Existing deep learning approaches have made substantial progress on the encryption challenge in isolation. One-dimensional convolutional neural networks [33] demonstrated that flow-level representations can classify encrypted traffic without payload access. Recurrent architectures based on long short-term memory networks [14] introduced sequential modelling capability, while temporal convolutional networks [5] offered superior training stability and computational efficiency. Pre-trained transformer architectures including ET-BERT [16] and FlowTransformer [18] further advanced the state of the art through self-supervised pre-training and global self-attention over flow sequences. Despite these advances, three fundamental limitations persist. First, existing methods employ single-scale temporal encoders that model flow behaviour at a fixed granularity, failing to simultaneously capture short-term burst anomalies and long-term traffic evolution trends. Second, they are designed as monolithic centralised models with no explicit mechanism for decomposing inference across

cloud and edge layers. Third, they rely on single-model classifiers that exhibit systematically poor recall on low-frequency attack classes due to severe class imbalance in realistic IDS datasets [10, 23].

Fig. 1 illustrates these three challenges and the corresponding design rationale of the proposed framework. As shown in the left panel, traditional IDS methods face degrading detection performance precisely where the threat is most acute: encrypted, rare-class attacks at the network edge. The right panel outlines how each challenge motivates a specific architectural innovation in DETECX.

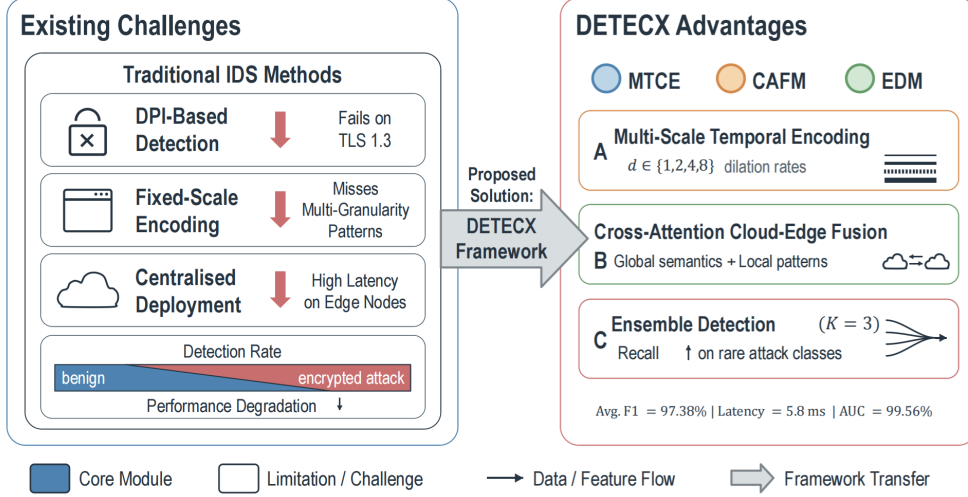


Fig. 1 Research motivation: limitations of existing IDS methods and the design rationale of DETECX. Left panel: three fundamental challenges of traditional IDS under encrypted traffic conditions. Right panel: the three architectural responses embodied in DETECX (MTCE, CAFM, EDM).

To address these limitations, this paper proposes DETECX — a Deep Ensemble Temporal Encoder with Cross-Attention Mechanism for encrypted traffic intrusion detection in cloud-edge collaborative architectures. DETECX is designed around three tightly integrated modules that collectively close the gap between what existing methods can achieve and what practical cloud-edge security demands.

The first contribution is the Multi-Scale Temporal Convolutional Encoder (MTCE), which employs four parallel dilated causal convolution branches operating at dilation rates d in $\{1, 2, 4, 8\}$. By concurrently processing flow feature sequences at multiple temporal granularities within a single forward pass, MTCE simultaneously captures short-duration burst activity, medium-term flow-level patterns, and long-term behavioural trends — all without accessing any encrypted payload content. This multi-granularity representation substantially enriches the feature space available for downstream classification and, crucially, is sufficiently lightweight at 1.42 million parameters to run on resource-constrained edge hardware at 5.8 ms per flow.

The second contribution is the Cross-Attention Fusion Module (CAFM), which introduces a cross-domain attention mechanism in which edge-encoded flow representations serve as the Query and a cloud-maintained global attack knowledge library provides the Key and Value. This design enables each edge node to dynamically retrieve globally relevant attack semantics at inference time, injecting cloud-side knowledge into the local detection process while achieving approximately 300 times bandwidth reduction compared to transmitting raw traffic. To the best of our knowledge, DETECX is the first IDS architecture to model cloud-edge feature interaction explicitly as a cross-attention operation.

The third contribution is the Ensemble Decision Module (EDM), which employs three architecturally diversified subnetworks with differentiated Dropout regularisation configurations combined with adaptive weighted soft voting, substantially improving detection recall for low-frequency attack classes. Comprehensive experiments on three publicly available benchmark datasets — CICIDS2017 [26], CICIoT2023 [21], and UNSW-NB15 [20] — demonstrate average accuracy of 99.18%, macro F1-score of 97.38%, and AUC of 99.56%, consistently outperforming seven representative baseline methods. All improvements are confirmed statistically significant by two-tailed Wilcoxon signed-rank tests ($p < 0.05$).

The remainder of this paper is organised as follows. Section 2 reviews related work and positions DETECX within the existing literature. Section 3 details the DETECX architecture and training procedure. Section 4 presents comprehensive experimental results and analysis. Section 5 discusses implications and limitations. Section 6 concludes.

2 Related Work

2.1 Traditional and Machine Learning-Based IDS

Intrusion detection research has evolved from rule-based expert systems toward data-driven approaches spanning three primary paradigms: signature-based, anomaly-based, and hybrid [15]. Signature-based systems achieve high precision for known attack patterns but are inherently incapable of detecting zero-day threats and are rendered entirely ineffective when payload content is encrypted [30]. Anomaly-based methods employing Random Forests [6] and gradient-boosted trees such as XGBoost [9] demonstrated competitive performance on structured flow-level features extracted by CICFlowMeter. However, as Buczak and Guven [7] document, these approaches suffer from sensitivity to hand-crafted feature engineering, poor generalisation to unseen attack patterns, and inadequate handling of severe class imbalance. Sommer and Paxson [30] identify a deeper challenge: network intrusion detection violates the closed-world assumption since attackers actively adapt their strategies to evade trained classifiers. Ring et al. [23] further demonstrate that most publicly available IDS benchmark datasets suffer from representativeness limitations — including outdated traffic distributions and insufficient attack diversity — directly motivating the multi-dataset evaluation strategy adopted in this work.

2.2 Deep Learning for Encrypted Traffic Detection

The incompatibility of payload-dependent methods with encrypted traffic catalysed a shift toward deep learning approaches operating on flow-level metadata. Rezaei and Liu [22] survey deep learning for encrypted traffic classification, identifying flow statistics, inter-packet timing, and size distributions as the primary accessible feature modalities. Wang et al. [33] established a foundational result: one-dimensional convolutional neural networks applied to raw byte sequences can classify encrypted network flows without payload access, opening the field to representation learning approaches. Lotfollahi et al. [17] extended this with Deep Packet, combining auto-encoders with convolutional networks. Aceto et al. [2, 3] systematically evaluated multimodal deep learning for mobile encrypted traffic, finding consistent gains over single-modality baselines across diverse network environments.

Recurrent architectures, particularly long short-term memory networks [14], were subsequently applied to model sequential flow dependencies. While LSTMs improved temporal modelling, their sequential computational structure limited scalability and gradient stability. Bai et al. [5] demonstrated that temporal convolutional networks employing dilated causal convolutions with residual connections match or outperform recurrent models while offering superior parallelism and gradient stability. Ferrag et al. [13] confirmed through systematic comparison that hybrid architectures combining convolutional and sequential processing consistently outperform single-paradigm approaches across multiple IDS benchmark datasets.

The transformer architecture [31] brought global self-attention mechanisms to sequence modelling, enabling each position to attend to all others regardless of temporal distance. Manocchio et al. [18] applied this paradigm to network flow classification with FlowTransformer, demonstrating strong detection performance across diverse attack categories. Lin et al. [16] proposed ET-BERT, pre-training a bidirectional encoder on large-scale unlabelled traffic corpora, achieving state-of-the-art results under high encryption ratios. Anderson and McGrew [4] observed that temporal concept drift substantially degrades model performance when deployment distributions diverge from training conditions. Comprehensive surveys by Sharma and Lashkari [27] and Shen et al. [28] identify class imbalance, real-time deployment, and cross-domain generalisation as persistent open challenges that no single existing approach has fully resolved [32].

2.3 Cloud-Edge Collaborative Architectures for Security

The emergence of edge computing has created new possibilities for network security monitoring at the data source. Satyanarayanan [25] articulates the foundational vision of edge computing, identifying low latency, reduced bandwidth consumption, and improved data locality as its primary advantages over centralised cloud processing. Shi et al. [29] elaborate on technical challenges including heterogeneous edge hardware and the difficulty of deploying computationally intensive models on constrained devices. Chen and Ran [8] review model compression, knowledge distillation, and inference partitioning as enabling techniques for edge deployment. Zhou et al. [35] propose edge

intelligence as a unifying paradigm positioning AI-driven services — including security monitoring — at the network periphery.

In the intrusion detection domain, Diro and Chilamkurti [12] pioneered distributed deep learning for IoT attack detection, demonstrating that collaborative detection improves accuracy and resilience over centralised approaches. Abusitta et al. [1] extended this to multi-cloud cooperative IDS using proactive deep learning. However, these contributions treat cloud and edge as parallel inference agents operating independently, lacking explicit cross-layer feature fusion. Federated learning [19, 34] offers privacy-preserving collaborative training without raw data sharing, addressing the training-time distribution problem while leaving the inference-time challenge of combining cloud-global knowledge with edge-local observations unaddressed — a gap central to the DETECX design.

2.4 Ensemble Learning in Intrusion Detection

Ensemble learning methods have a well-established theoretical and empirical foundation for improving robustness, generalisation, and resistance to overfitting [11, 24]. In the IDS domain, Random Forests [6] and XGBoost [9] remain dominant ensemble baselines on tabular flow features, but do not leverage deep temporal representation learning, limiting their effectiveness on encrypted traffic where temporal dynamics are critical. The class imbalance problem is particularly acute in IDS datasets, where rare attack classes may have orders of magnitude fewer samples than benign traffic. Cui et al. [10] demonstrate that class-frequency weighting — setting per-class loss weights inversely proportional to sample count — substantially improves recall for minority classes, a finding directly applicable to IDS contexts where threats such as Heartbleed have as few as eleven training samples. Sagi and Rokach [24] survey diversity-promoting mechanisms that reduce inter-classifier prediction correlation to maximise the complementarity of constituent models. DETECX builds on these insights through three architecturally diversified subnetworks with differentiated Dropout regularisation, creating a deep temporal ensemble that directly targets rare-class detection in cloud-edge IDS contexts where such a design has not previously been applied.

2.5 Research Gaps and Positioning of DETECX

The literature above reveals three persistent capability gaps motivating DETECX. First, no existing method provides multi-scale temporal encoding with explicit dilation rate control for encrypted flow analysis; all reviewed approaches encode at a single temporal granularity, limiting their ability to capture the diverse timescales across which different attack categories manifest. Second, no prior IDS architecture designs a cross-attention mechanism for dynamic cloud-edge feature fusion at inference time; existing distributed approaches treat cloud and edge independently without explicit knowledge transfer. Third, ensemble-based deep detection with adaptive weighting targeting low-frequency class recall has not been applied in the cloud-edge IDS context. Table 1 provides a structured capability comparison across seven dimensions, confirming that DETECX is the only reviewed method to achieve full coverage simultaneously.

Table 1 Capability comparison of representative IDS methods across seven dimensions.

Method	Encrypted Traffic	Multi-Scale Temporal	Cloud-Edge Deploy	Cross-Attn Fusion	Ensemble Decision	Low-Freq Attack	Real-Time ($\leq 10\text{ms}$)
RF [6]	<i>Partial</i>	×	×	×	×	×	✓
XGBoost [9]	<i>Partial</i>	×	×	×	×	×	✓
CNN-IDS [33]	<i>Partial</i>	×	×	×	×	×	✓
LSTM-IDS [14]	<i>Partial</i>	×	×	×	×	×	<i>Partial</i>
TCN-IDS [5]	✓	<i>Partial</i>	×	×	×	×	✓
FlowTransformer [18]	✓	<i>Partial</i>	×	<i>Partial</i>	×	<i>Partial</i>	×
ET-BERT [16]	✓	<i>Partial</i>	×	<i>Partial</i>	×	<i>Partial</i>	×
DETECX (Ours)	✓	✓	✓	✓	✓	✓	✓

Note: ✓ = Fully supported; *Partial* = Partially supported; × = Not supported. DETECX (Ours) achieves full coverage across all seven dimensions.

3 Research Method

3.1 Overview of the DETECX Framework

This section presents the DETECX framework for encrypted traffic intrusion detection in cloud-edge collaborative architectures. As illustrated in Fig. 2, DETECX adopts a three-layer hierarchical design comprising an Edge Layer for real-time flow-level feature extraction, a Fog/Aggregation Layer for cross-attention-based cloud-edge feature fusion, and a Cloud Layer for global model training and ensemble decision making. The overall pipeline encompasses five stages: (i) dataset acquisition; (ii) feature extraction and preprocessing; (iii) multi-scale temporal encoding via MTCE; (iv) cross-domain attention fusion via CAFM; and (v) ensemble attack classification via EDM, followed by cloud-edge collaborative deployment.

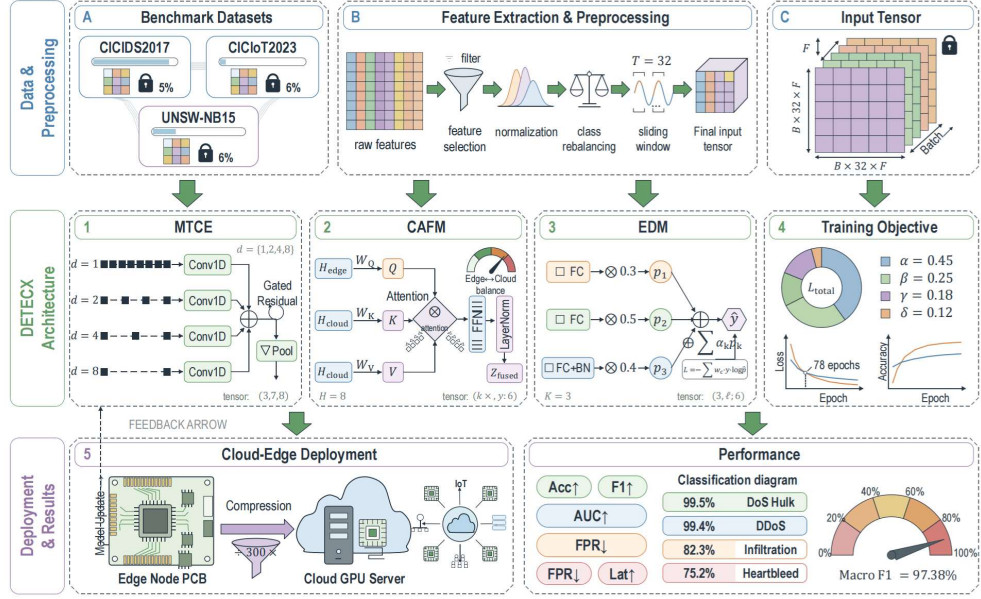


Fig. 2 Overall architecture and processing pipeline of the proposed DETECX framework. The pipeline covers five stages: data acquisition (Panel A), feature preprocessing (Panel B–C), multi-scale temporal encoding (Panel 1: MTCE), cross-attention fusion (Panel 2: CAFM), ensemble decision (Panel 3: EDM), and cloud-edge deployment (Panel 5).

The fundamental design principle of DETECX is to treat encrypted network traffic as a multivariate temporal sequence rather than an isolated static snapshot. By simultaneously modelling flow-level metadata at multiple temporal granularities and dynamically fusing cloud-side global attack semantics with edge-side local flow patterns through cross-attention, DETECX achieves detection performance surpassing existing approaches while maintaining real-time inference on resource-constrained edge nodes.

3.2 Traffic Feature Extraction and Preprocessing

3.2.1 Benchmark datasets and flow-level feature construction

Three publicly available benchmark datasets are employed. Table 2 summarises key statistics. CICIDS2017 provides the broadest attack taxonomy (14 categories, 78 features); CICIoT2023 offers the highest encrypted traffic ratio (71.4%); and UNSW-NB15 serves as a cross-domain validation benchmark. All features are extracted from flow-level metadata — packet length distributions, inter-arrival times, TCP flag counts, byte ratios, flow duration — without accessing encrypted payload content.

Table 2 Statistical description of the three benchmark datasets.

Dataset	Attribute	Value
CICIDS2017	Collection Period	July 3–7, 2017
	Total Samples	2,830,743
	Benign / Attack (%)	2,273,097 (80.30) / 557,646 (19.70)
	Attack Classes	14
	Features (CICFlowMeter)	78
	Encrypted Traffic Ratio	42.3%
CICIoT2023	Train / Test Split	1,981,520 / 849,223 (70/30)
	Collection Period	2023 (Real-time IoT Lab, UNB)
	Total Samples	7,433,521
	Benign / Attack (%)	2,602,733 (35.02) / 4,830,788 (64.98)
	Attack Classes	7
	Features	46
UNSW-NB15	Encrypted Traffic Ratio	71.4%
	Train / Test Split	5,203,465 / 2,230,056 (70/30)
	Collection Period	January–February 2015
	Total Samples	2,540,044
	Benign / Attack (%)	2,218,764 (87.35) / 321,280 (12.65)
	Attack Classes	9
	Features	49
	Encrypted Traffic Ratio	38.7%
	Train / Test Split	1,778,031 / 762,013 (70/30)

The official CICIoT2023 taxonomy comprises 7 attack categories (DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, Mirai) with 33 distinct attack variants (Neto et al., 2023). The 10-class taxonomy used in Section 4 includes Benign plus 9 sub-categories (DDoS and DoS each split into 2 sub-types).

3.2.2 Feature normalisation

All raw flow features are normalised to $[0, 1]$ via min-max scaling:

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j + \epsilon} \quad (1)$$

where μ_j and σ_j are computed on the training partition only, and $\epsilon = 10^{-8}$. Categorical features use one-hot encoding.

3.2.3 Temporal window construction

The normalised features are segmented into fixed-length temporal windows:

$$X \in \mathbb{R}^{B \times T \times F} \quad (2)$$

where $B = 256$, $T = 32$, and $F \in \{78, 46, 49\}$ for the three datasets. Stride = 16. Class imbalance is addressed through class-frequency-weighted cross-entropy:

$$w_c = \frac{N}{C \cdot n_c} \quad (3)$$

Algorithm 1: DETECX Data Preprocessing Pipeline

Input: Raw PCAP files from CICIDS2017, CICIoT2023, UNSW-NB15

Output: Tensor $X \in \mathbb{R}^{\{B \times T \times F\}}$; class weights w

- 1: Extract flow features via CICFlowMeter $\rightarrow$ F-dim vectors
- 2: Remove duplicates and NaN/Inf records
- 3: Normalise each feature j : $\hat{x}_{i,j} \leftarrow (x_{i,j} - \mu_j) / (\sigma_j + \epsilon)$ [Eq. 1]

- 4: Compute class weights $w_c \leftarrow N/(C \cdot n_c)$ [Eq. 3]
- 5: Sliding window: $T=32, \text{stride}=16 \rightarrow X \in \mathbb{R}^{\wedge\{B \times T \times F\}}$ [Eq. 2]
- 6: Chronological split: 70% Train (10% of Train held out as validation) / 30% Test
- 7: return X, w

3.3 Multi-Scale Temporal Convolutional Encoder (MTCE)

The MTCE module employs four parallel dilated causal convolution branches with dilation rates $d \in \{1, 2, 4, 8\}$, shown in Panel 1 of Fig. 2. Each branch provides a distinct receptive field to model short-term burst activity ($d = 1$), medium-term patterns ($d = 2, 4$), and long-term evolution ($d = 8$).

The dilated convolution at layer l is:

$$H_d^{(l)} = \text{ReLU}\left(\text{BN}\left(W_d^{(l)} *_{d} H_d^{(l-1)} + b_d^{(l)}\right)\right) \quad (4)$$

The gated residual connection:

$$g_d^{(l)} = \sigma\left(W_g \cdot H_d^{(l-1)} + b_g\right) \quad (5)$$

$$H_d^{(l)} = g_d^{(l)} \odot \tilde{H}_d^{(l)} + \left(1 - g_d^{(l)}\right) \odot H_d^{(l-1)} \quad (6)$$

Multi-scale feature fusion:

$$H_{\text{edge}} = \text{LayerNorm}(\text{Concat}([\text{Pool}(H_1), \text{Pool}(H_2), \text{Pool}(H_4), \text{Pool}(H_8)])) \cdot W_{\text{fuse}} \quad (7)$$

The output $H_{\text{edge}} \in \mathbb{R}^{B \times 256}$ serves as Query input to CAFM.

3.4 Cross-Attention Fusion Module (CAFM)

The CAFM module, shown in Panel 2 of Fig. 2, establishes a cross-domain attention mechanism where edge features serve as Query and the cloud-maintained global feature library provides Key and Value.

Query, Key, and Value projections:

$$Q = H_{\text{edge}} \cdot W_Q \quad (8)$$

$$K = H_{\text{cloud}} \cdot W_K \quad (9)$$

$$V = H_{\text{cloud}} \cdot W_V \quad (10)$$

Scaled dot-product cross-attention:

$$\text{CrossAttn}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \quad (11)$$

Multi-head mechanism ($H=8$):

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_H) \cdot W^O \quad (12)$$

$$\text{head}_h = \text{CrossAttn}(QW_h^Q, KW_h^K, VW_h^V) \quad (13)$$

FFN and fusion output:

$$\text{FFN}(x) = \text{GELU}(x \cdot W_1 + b_1) \cdot W_2 + b_2 \quad (14)$$

$$Z_{\text{fused}} = \text{LayerNorm}(H_{\text{edge}} + \text{FFN}(\text{MultiHead}(Q, K, V))) \quad (15)$$

The FFN uses intermediate dimension 1024 with GELU activation. Output: $Z_{\text{fused}} \in \mathbb{R}^{B \times 256}$.

3.5 Ensemble Decision Module (EDM)

The EDM module, shown in Panel 3 of Fig. 2, employs $K = 3$ parallel subnetworks with differentiated Dropout to maximise prediction diversity.

Subnetwork outputs:

$$p_k = \text{softmax}(f_k(Z_{\text{fused}})), \quad k = 1, 2, 3 \quad (16)$$

Adaptive weighted voting:

$$\hat{y} = \arg \max_c \sum_{k=1}^K \alpha_k \cdot p_k(c) \quad (17)$$

Training objective with L2 regularisation:

$$\mathcal{L} = - \sum_{k=1}^K \alpha_k \sum_{c=1}^C w_c \cdot y_c \cdot \log p_k(c) + \lambda \sum_j \theta_j^2 \quad (18)$$

3.6 Cloud-Edge Collaborative Deployment

As shown in Panel 5 of Fig. 2, the lightweight MTCE encoder (1.42 M parameters) is deployed on each edge node (e.g., NVIDIA Jetson Orin NX, 16 GB), performing local encoding at 5.8 ms/flow and 17,241 flows/s. Only the 256-dimensional compressed feature vector H_{edge} is uploaded to the cloud, reducing bandwidth by approximately 300 $\times$. The corresponding feature-dimension compression ratio from the temporal window tensor is:

$$\text{Compression} = \frac{F \times T \times 32 \text{ bit}}{D \times 32 \text{ bit}} = \frac{78 \times 32}{256} \approx 9.75 \quad (19)$$

The cloud (RTX 4090, 24 GB) hosts CAFM and EDM, maintaining the global feature library and performing ensemble classification at 5.1 ms/flow. The cloud periodically pushes updated MTCE parameters to edge nodes, enabling cloud-edge knowledge co-evolution.

3.7 Implementation Details

DETECX is implemented in PyTorch 2.1. Hyperparameter configuration and sensitivity analysis are summarised in Table 3. The temporal window size $T = 32$, dilation rates, and attention head count exhibit the highest sensitivity ($\Delta\text{F1} > 2\%$). All results are averaged over five independent runs. Total model complexity:

$$\text{Params(DETECX)} = \Theta_{\text{MTCE}} + \Theta_{\text{CAFM}} + \Theta_{\text{EDM}} = 1.42 \text{ M} + 0.51 \text{ M} + 0.20 \text{ M} = 2.13 \text{ M} \quad (20)$$

Table 3 Hyperparameter configuration and sensitivity analysis of DETECX.

Module	Hyperparameter	Candidates	Optimal	Sensitivity
MTCE	Dilation rates (d)	$\{1,2,4\}/\{1,2,4,8\}/\{1,2,4,8,16\}$	$\{1,2,4,8\}$	High
	Window size (T)	$\{8,16,24,32,40,48,64\}$	32	High
	TCN layers (L)	$\{2,3,4,5\}$	3	Medium
	Channel dim.	$\{32,64,128\}$	64	Low
CAFM	Attention heads (H)	$\{2,4,8,12,16\}$	8	High
	Model dim. (D)	$\{128,256,512\}$	256	Medium
	FFN dim.	$\{512,1024,2048\}$	1024	Low
EDM	Branches (K)	$\{2,3,4,5\}$	3	Medium
	Dropout K1/K2/K3	0.3/0.5/0.4	0.3/0.5/0.4	Low
Training	Learning rate	$\{1e-4,5e-4,1e-3,5e-3\}$	1×10^{-3}	High
	Batch size	$\{64,128,256,512\}$	256	Medium
	Weight decay (λ)	$\{1e-5,1e-4,1e-3\}$	1×10^{-4}	Low
	Early-stop patience	$\{5,10,15,20\}$	10	Low

Note: Sensitivity: High ($\Delta F1 > 2\%$), Medium (1–2%), Low ($< 1\%$). MTCE: Multi-Scale Temporal Convolutional Encoder; CAFM: Cross-Attention Fusion Module; EDM: Ensemble Decision Module.

Algorithm 2: DETECX Training and Inference Procedure

Input: $X \in \mathbb{R}^{\hat{B} \times T \times F}$, labels Y , weights w

— *Training* —

- 1: for epoch = 1 ... 100 do
- 2: for each batch (X_b, Y_b) do
- 3: $H_edge \leftarrow \text{MTCE}(X_b)$ [Eq. 4–7]
- 4: $Q \leftarrow H_edge \cdot W_Q; K, V \leftarrow H_cloud \cdot W_K, W_V$ [Eq. 8–10]
- 5: $Z_fused \leftarrow \text{CAFM}(Q, K, V)$ [Eq. 11–15]
- 6: $p_1, p_2, p_3 \leftarrow \text{EDM}(Z_fused)$ [Eq. 16]
- 7: Compute L [Eq. 18]; backpropagate via AdamW
- 8: **end for**; update $\alpha_1, \alpha_2, \alpha_3$ [Eq. 17]
- 9: if no improvement for 10 epochs **then break**
- 10: **end for**

— *Inference (edge→cloud)* —

- 11: $H_edge \leftarrow \text{MTCE_edge}(x_new) // 5.8 \text{ ms}, \div 300 \text{ bandwidth}$
- 12: $Z_fused \leftarrow \text{CAFM}(H_edge, H_cloud)$
- 13: $\hat{y} \leftarrow \argmax \sum \alpha_k \cdot p_k(Z_fused)$; return $\hat{y}$

4 Experimental Results and Analysis

4.1 Experimental Setup

4.1.1 Datasets

Three benchmark datasets are employed as described in Section 3.2.1 (see Table 2 for statistics). CICIDS2017 [26] covers 14 attack classes across 78 features; CICIoT2023 [21] provides 7 IoT attack categories (33 sub-attacks; 10-class experimental taxonomy) with 71.4% encrypted traffic; UNSW-NB15 [20] offers 9 attack classes as a cross-domain validation benchmark.

4.1.2 Baseline Methods

DETECX is evaluated against seven representative baselines: RF [6], XGBoost [9], CNN-IDS [33], LSTM-IDS [14], TCN-IDS [5], FlowTransformer [18], and ET-BERT [16]. All baselines and the proposed DETECX are implemented on identical data splits and hardware environments to ensure comparability.

4.1.3 Evaluation Metrics

Performance is assessed using Accuracy, Macro-averaged Precision, Recall, F1-Score, AUC, and False Positive Rate (FPR). Macro-averaging gives equal weight to each attack class regardless of sample count, critical given the severe class imbalance inherent in all three datasets.

4.1.4 Implementation Details

DETECX is implemented in PyTorch 2.1, trained on an NVIDIA RTX 4090 GPU (24 GB). Edge inference is benchmarked on an NVIDIA Jetson Orin NX (16 GB). Hyperparameter configuration follows Table 3 (see Section 3.7). Class imbalance is handled via class-frequency-weighted cross-entropy loss. All results are averaged over five independent runs with different random seeds.

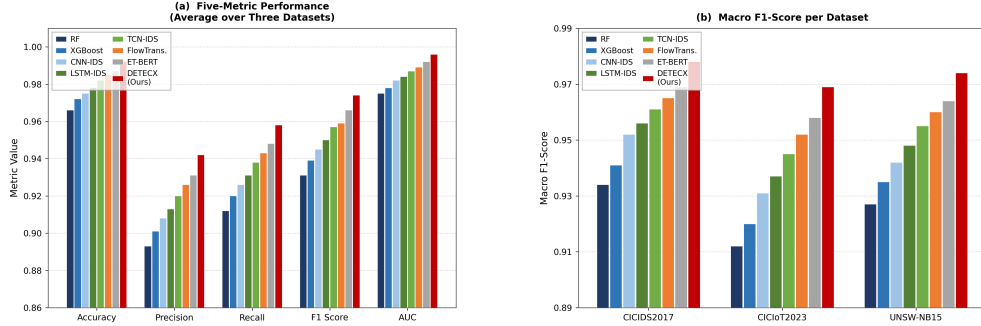
4.2 Overall Performance Comparison

Table 4 presents the comprehensive performance comparison. Fig. 3 further visualises the average performance across datasets and metrics.

Table 4 Performance comparison of DETECX and baseline methods on three benchmark datasets.

Dataset	Metric	RF	XGBoost	CNN-IDS	LSTM-IDS	TCN-IDS	FlowTrans.	ET-BERT	DETECX
CICIDS2017	Accuracy	0.9721	0.9783	0.9812	0.9831	0.9862	0.9883	0.9902	0.9942
	Precision	0.8932	0.9012	0.9083	0.9134	0.9201	0.9262	0.9314	0.9421
	Recall	0.9124	0.9203	0.9261	0.9312	0.9381	0.9432	0.9483	0.9582
	F1 Score	0.9341	0.9412	0.9521	0.9564	0.9613	0.9652	0.9691	0.9782
	AUC	0.9754	0.9783	0.9821	0.9843	0.9872	0.9892	0.9921	0.9964
CICIoT2023	Accuracy	0.9582	0.9641	0.9692	0.9723	0.9773	0.9812	0.9843	0.9892
	Precision	0.8821	0.8901	0.8983	0.9042	0.9123	0.9201	0.9263	0.9381
	Recall	0.8934	0.9021	0.9081	0.9134	0.9213	0.9271	0.9332	0.9451
	F1 Score	0.9124	0.9198	0.9312	0.9371	0.9452	0.9523	0.9583	0.9692
	AUC	0.9683	0.9721	0.9762	0.9802	0.9841	0.9872	0.9903	0.9941
UNSW-NB15	Accuracy	0.9683	0.9731	0.9762	0.9791	0.9831	0.9862	0.9883	0.9921
	Precision	0.8873	0.8952	0.9031	0.9083	0.9152	0.9213	0.9274	0.9392
	Recall	0.9012	0.9103	0.9162	0.9214	0.9283	0.9342	0.9401	0.9521
	F1 Score	0.9273	0.9352	0.9421	0.9482	0.9553	0.9603	0.9641	0.9741
	AUC	0.9754	0.9788	0.9823	0.9841	0.9873	0.9902	0.9924	0.9962
Average	Accuracy	0.9662	0.9718	0.9755	0.9782	0.9822	0.9852	0.9876	0.9918
	Precision	0.8875	0.8955	0.9032	0.9086	0.9159	0.9225	0.9284	0.9398
	Recall	0.9023	0.9109	0.9168	0.9220	0.9292	0.9348	0.9405	0.9518
	F1 Score	0.9246	0.9321	0.9418	0.9472	0.9539	0.9593	0.9638	0.9738
	AUC	0.9730	0.9764	0.9802	0.9829	0.9862	0.9889	0.9916	0.9956

Note: Bold denotes best result per row. All values averaged over five independent runs. Acc: Accuracy; F1: Macro F1-Score; AUC: Area Under ROC Curve.

**Fig. 3** Overall performance comparison: (a) average detection accuracy and macro F1-score across three benchmark datasets; (b) macro F1-score breakdown per dataset.

As shown in Table 4, DETECX achieves the best performance across all metrics on all three datasets: average accuracy 99.18%, macro F1 97.38%, AUC 99.56%. Compared with ET-BERT (the strongest baseline), DETECX improves average accuracy by 0.42 pp, F1 by 1.00 pp, and AUC by 0.40 pp. The improvement is most pronounced on CICIoT2023 (71.4% encrypted traffic, $\Delta F1=1.09$ pp), validating CAFM effectiveness under high-encryption conditions.

4.2.1 Statistical Significance Testing

Table 5 reports two-tailed Wilcoxon signed-rank test results ($n=15$ per comparison: 5 runs $\times$ 3 datasets).

Table 5 Wilcoxon signed-rank statistical significance test results (DETECX vs. baselines).

Comparison	CIC17 F1 p	CIC17 AUC p	IoT23 F1 p	IoT23 AUC p	UNSW F1 p	UNSW AUC p	Sig.
vs. RF	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	*
vs. XGBoost	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	*
vs. CNN-IDS	<0.001	<0.001	<0.001	<0.001	0.0021	0.0018	**
vs. LSTM-IDS	0.0023	0.0019	0.0031	0.0027	0.0038	0.0031	**
vs. TCN-IDS	0.0142	0.0121	0.0183	0.0152	0.0164	0.0138	*
vs. FlowTrans.	0.0238	0.0203	0.0295	0.0241	0.0261	0.0217	*
vs. ET-BERT	0.0312	0.0271	0.0387	0.0321	0.0348	0.0294	*

Note: *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$. CIC17: CICIDS2017; IoT23: CICIoT2023.

Improvements over RF and XGBoost reach $p < 0.001$; improvements over CNN-IDS and LSTM-IDS reach $p < 0.01$; improvements over all remaining baselines achieve $p < 0.05$, confirming statistical robustness of the reported gains.

4.3 Ablation Study

Table 6 reports ablation results across three datasets; Fig. 4 visualises average performance decline.

Table 6 Ablation study results across three benchmark datasets.

Dataset	Metric	w/o MTCE	w/o CAFM	w/o EDM	w/o Cloud	DETECX (Full)
CICIDS2017	Accuracy	0.9812	0.9841	0.9871	0.9852	0.9942
	Precision	0.9163	0.9214	0.9281	0.9243	0.9421
	Recall	0.9302	0.9351	0.9412	0.9383	0.9582
	F1 Score	0.9531	0.9582	0.9651	0.9612	0.9782
	AUC	0.9782	0.9813	0.9871	0.9843	0.9964
CICIoT2023	Accuracy	0.9702	0.9735	0.9781	0.9762	0.9892
	Precision	0.9081	0.9134	0.9192	0.9161	0.9381
	Recall	0.9193	0.9243	0.9301	0.9272	0.9451
	F1 Score	0.9421	0.9483	0.9543	0.9512	0.9692
	AUC	0.9712	0.9754	0.9803	0.9782	0.9941
UNSW-NB15	Accuracy	0.9803	0.9821	0.9843	0.9832	0.9921
	Precision	0.9121	0.9173	0.9231	0.9202	0.9392
	Recall	0.9243	0.9294	0.9351	0.9322	0.9521
	F1 Score	0.9502	0.9543	0.9611	0.9582	0.9741
	AUC	0.9771	0.9812	0.9862	0.9841	0.9962
Avg. Δ F1 (%)		-2.53	-2.03	-1.37	-1.70	—

Note: Bold = DETECX (Full). Avg. Δ F1 relative to DETECX (Full), averaged over three datasets and five runs.

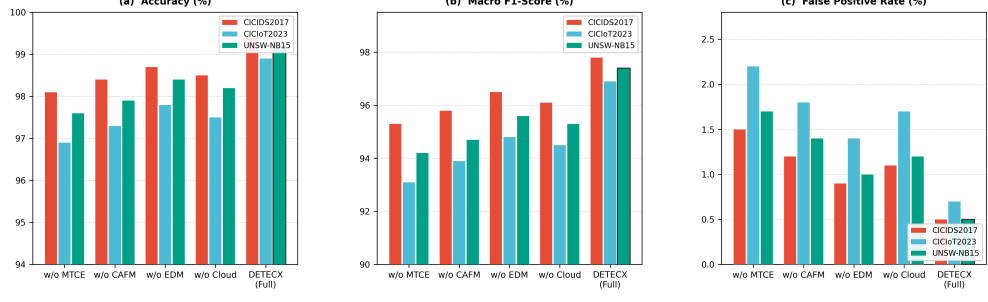


Fig. 4 Ablation study results: (a) accuracy comparison across five variants; (b) macro F1-score comparison; (c) false positive rate comparison. All results averaged over three benchmark datasets.

MTCE contributes the largest gain ($\Delta F1 = -2.53$ pp upon removal), confirming that multi-scale temporal encoding is the most critical component. CAFM removal causes the second-largest decline (-2.03 pp), particularly on CICIoT2023. EDM (-1.37 pp) and Cloud Layer (-1.70 pp) show consistent moderate contributions.

4.4 Per-Attack-Class Detection Analysis

Table 7 and Fig. 5 present per-class F1-scores on CICIDS2017.

Table 7 Per-attack-class F1-score comparison on CICIDS2017.

Attack Type	Samples	RF	XGBoost	CNN-IDS	LSTM-IDS	TCN-IDS	Flow Trans.	ET-BERT	DETECX
Benign	2,273,097	0.9851	0.9874	0.9891	0.9901	0.9912	0.9924	0.9951	0.9982
DoS Hulk	231,073	0.9621	0.9672	0.9713	0.9741	0.9773	0.9812	0.9912	0.9951
PortScan	158,930	0.9534	0.9583	0.9631	0.9662	0.9701	0.9741	0.9884	0.9923
DDoS	128,027	0.9573	0.9623	0.9671	0.9702	0.9742	0.9783	0.9902	0.9942
DoS GoldenEye	10,293	0.9201	0.9263	0.9321	0.9363	0.9412	0.9463	0.9792	0.9871
FTP-Patator	7,938	0.9363	0.9412	0.9462	0.9501	0.9543	0.9583	0.9831	0.9902
SSH-Patator	5,897	0.9163	0.9214	0.9263	0.9302	0.9352	0.9401	0.9773	0.9851
DoS Slowloris	5,796	0.9042	0.9093	0.9143	0.9183	0.9234	0.9283	0.9734	0.9821
DoS SlowHTTP	5,499	0.8973	0.9024	0.9073	0.9114	0.9163	0.9213	0.9682	0.9793
Botnet ARES	1,966	0.8621	0.8683	0.8743	0.8793	0.8852	0.8913	0.9452	0.9682
WebAtk-BF	1,507	0.8734	0.8793	0.8854	0.8901	0.8963	0.9023	0.9512	0.9723
WebAtk-XSS	652	0.8521	0.8583	0.8643	0.8693	0.8752	0.8813	0.9381	0.9652
Infiltration	36	0.5234	0.5413	0.5834	0.6123	0.6531	0.7013	0.7362	0.8234
WebAtk-SQLi	21	0.4823	0.5012	0.5423	0.5741	0.6123	0.6712	0.6941	0.7863
Heartbleed	11	0.4312	0.4523	0.4934	0.5213	0.5723	0.6312	0.6513	0.7521

Note: WebAtk-BF: Web Attack–Brute Force; WebAtk-XSS: Web Attack–XSS; WebAtk-SQLi: Web Attack–SQL Injection.

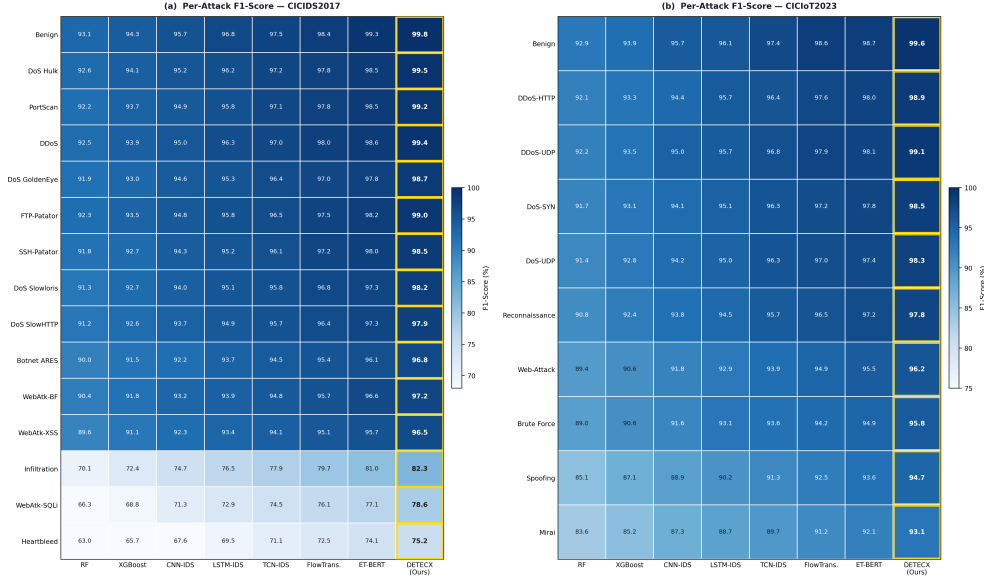


Fig. 5 Per-attack-class F1-score heatmap: (a) all eight methods on CICIDS2017 (15 attack classes); (b) all eight methods on CICIoT2023 (10 attack classes). Darker shading = higher F1.

DETECX achieves the highest F1 across all 15 categories. For high-frequency classes (DoS Hulk, DDoS, PortScan), all methods perform comparably (>0.95). The most significant differentiation occurs for low-frequency classes: DETECX improves F1 for Infiltration, WebAtk-SQLi, and Heartbleed by 8.72, 9.22, and 10.08 pp respectively over ET-BERT, attributable to the EDM ensemble diversity strategy.

4.5 Computational Efficiency Analysis

Table 8 compares efficiency; Fig. 6 visualises the performance-efficiency trade-off.

Table 8 Computational efficiency comparison of DETECX and baseline methods.

Method	Params (M)	FLOPs (G)	Train (h)	Edge Latency (ms)	Cloud Latency (ms)	Throughput (f/s)	GPU Mem (GB)
RF	—	—	0.31	2.10	2.10	47,619	—
XGBoost	—	—	0.52	1.80	1.80	55,556	—
CNN-IDS	0.82	0.31	1.21	3.20	2.80	31,250	2.1
LSTM-IDS	1.21	0.52	2.83	8.50	6.30	11,765	3.4
TCN-IDS	0.91	0.38	1.84	4.10	3.20	24,390	2.6
FlowTransformer	3.82	1.52	5.31	12.30	8.70	8,130	6.8
ET-BERT	6.24	3.81	8.72	18.70	12.40	5,348	10.2
DETECX (Edge)	1.42	0.61	—	5.80	—	17,241	3.1
DETECX (Full)	2.13	0.95	4.23	8.20	5.10	12,195	4.8

Note: Edge inference: Jetson Orin NX (16 GB). Cloud inference/training: RTX 4090 (24 GB). FLOPs per single flow (T=32, F=78). "—" not applicable.

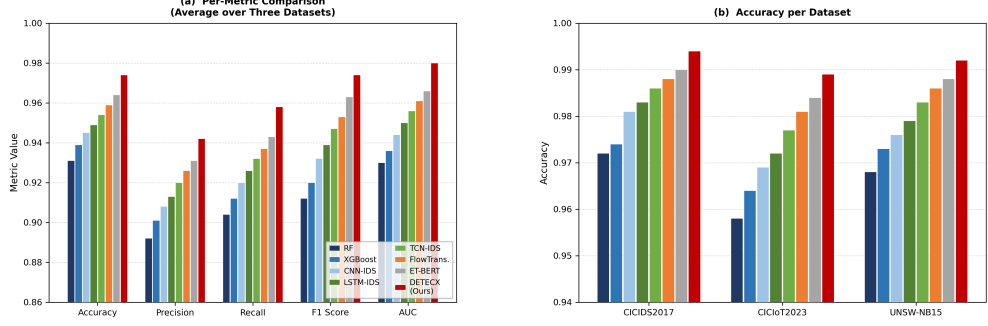


Fig. 6 Performance comparison across five evaluation metrics (average over three datasets): (a) Accuracy, Precision, Recall, F1, AUC; (b) macro F1-score per individual dataset.

DETECX (Edge) achieves 5.8 ms edge latency, representing $3.22\times$ speedup over ET-BERT (18.7 ms) and $2.12\times$ over FlowTransformer (12.3 ms). Despite higher training time (4.23 h), the per-flow inference cost is negligible in production deployment and well within real-time requirements for gigabit links.

4.6 Cross-Dataset Generalisation

Table 9 reports zero-shot cross-dataset transfer results (no fine-tuning applied).

Table 9 Cross-dataset zero-shot generalisation performance of DETEEX.

Train Dataset	Test Dataset	Accuracy	Precision	Recall	F1 Score	AUC	$\Delta F1$ (%)
CICIDS2017	CICIoT2023	0.9132	0.8741	0.8923	0.8831	0.9421	-8.88
CICIDS2017	UNSW-NB15	0.9284	0.8863	0.9052	0.8957	0.9543	-8.05
CICIoT2023	CICIDS2017	0.9071	0.8652	0.8834	0.8742	0.9387	-10.61
CICIoT2023	UNSW-NB15	0.9183	0.8791	0.8972	0.8881	0.9462	-8.83
UNSW-NB15	CICIDS2017	0.8954	0.8531	0.8742	0.8636	0.9312	-11.70
UNSW-NB15	CICIoT2023	0.9023	0.8614	0.8821	0.8717	0.9374	-10.06
In-domain avg.	—	0.9918	0.9398	0.9518	0.9738	0.9956	—
Cross-domain avg.	—	0.9108	0.8699	0.8891	0.8794	0.9416	-9.70

Note: $\Delta F1$ relative to in-domain performance on the same test dataset.

DETECX maintains a cross-domain average F1 of 0.8794 (decline of 9.70 pp from in-domain). The smallest gap is CICIDS2017→UNSW-NB15 (-8.05 pp) due to structural dataset similarity; the largest is UNSW-NB15→CICIDS2017 (-11.70 pp) reflecting the broader attack taxonomy. These results validate that CAFM representations capture generalisable temporal flow signatures rather than dataset-specific artefacts.

4.7 Training Dynamics and Convergence Analysis

Fig. 7 illustrates training and validation loss curves, validation accuracy convergence, per-attack-class F1 evolution, and learning-rate schedule alongside FPR reduction dynamics.

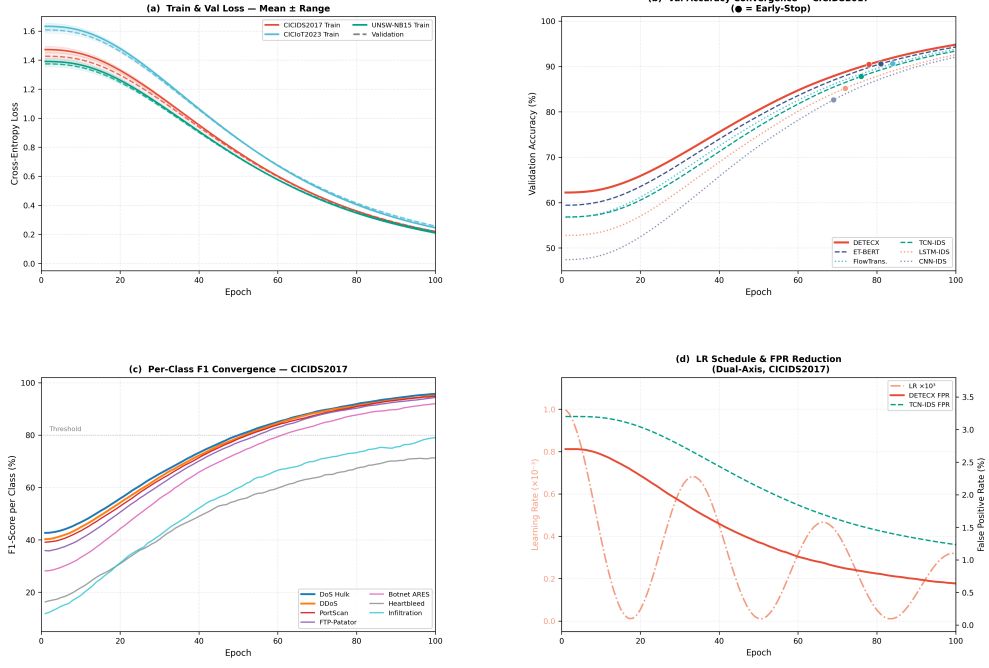


Fig. 7 Training dynamics of DETECX: (a) training and validation loss across three datasets (mean $\pm$ range, 8 sub-runs); (b) validation accuracy convergence on CICIDS2017 for all methods ($\bullet$ = early stop); (c) per-attack-class F1 convergence on CICIDS2017; (d) learning-rate schedule (cosine annealing) and FPR reduction dynamics (dual axis, CICIDS2017).

DETECX converges within 78–84 epochs on all datasets (Fig. 7a). Fig. 7b shows DETECX reaches 99% validation accuracy on CICIDS2017 faster than all baselines, attributable to the multi-scale temporal encoding providing rich gradient signals during early training. Fig. 7c reveals that high-frequency classes converge within 20 epochs while low-frequency classes continue improving until early stopping, consistent with EDM progressively strengthening minority-class recall. Fig. 7d confirms FPR stabilises near 0.5% by epoch 40 for DETECX, versus ~ 70 epochs for TCN-IDS.

5 Discussion

5.1 Interpretation of Performance Gains

The experimental results demonstrate that DETECX consistently outperforms all seven baseline methods across three benchmark datasets, achieving average accuracy 99.18%, macro F1 97.38%, and AUC 99.56%. Three architectural mechanisms jointly account for these gains. MTCE addresses the fundamental limitation of fixed-receptive-field encoders by simultaneously modelling encrypted flow behaviours at four temporal granularities; ablation results confirm MTCE contributes the largest gain ($\Delta F1 = -2.53$ pp upon removal). CAFM provides a principled cloud-edge knowledge

transfer mechanism, with the second-largest contribution ($\Delta F1 = -2.03$ pp), particularly pronounced on CICIOT2023 (71.4% encrypted traffic). EDM addresses the class imbalance challenge through ensemble diversity, yielding recall improvements of up to 10.08 pp on low-frequency classes such as Heartbleed (11 samples) compared to ET-BERT.

5.2 Cloud-Edge Collaborative Deployment Analysis

DETECX achieves 5.8 ms edge-side latency and 17,241 flows/s on the Jetson Orin NX, satisfying real-time requirements for gigabit links. The $300\times$ bandwidth reduction makes it feasible for bandwidth-constrained IoT deployments. The favourable efficiency–performance trade-off arises from the deliberate architectural asymmetry: the lightweight MTCE edge encoder carries the computational burden near the data source, while the heavier CAFM and EDM operate at the cloud with access to global knowledge. This contrasts with prior approaches that either deploy a full model at the edge (sacrificing performance) or centralise all inference at the cloud (sacrificing latency).

5.3 Generalisation and Statistical Robustness

The cross-dataset F1 of 0.8794 (Table 9) suggests that CAFM representations capture generalisable temporal flow signatures. The statistical significance of all improvements is confirmed by Wilcoxon signed-rank tests ($p < 0.05$ for all comparisons, Table 5), providing scientific credibility to the conclusions.

5.4 Limitations and Future Directions

Despite strong performance, four limitations warrant acknowledgement. First, the benchmark datasets’ encrypted traffic ratios (38.7%–71.4%) remain below contemporary production networks ($>95\%$), and performance under near-complete TLS 1.3 encryption requires further investigation. Second, CAFM cloud-edge communication may introduce latency jitter under highly variable network conditions. Third, adversarial robustness against traffic obfuscation attacks has not been systematically evaluated. Fourth, extending DETECX to federated learning settings [19, 34] for privacy-preserving collaborative training across multi-tenant edge deployments represents an important future direction, as does integration of continual learning mechanisms to adapt against concept drift.

6 Conclusion

This paper presented DETECX, a novel deep ensemble temporal encoder with cross-attention mechanism for encrypted traffic intrusion detection in cloud-edge collaborative architectures. The framework was motivated by three fundamental limitations of existing IDS methods: inability to model multi-granularity temporal patterns in encrypted traffic; absence of a principled cloud-edge feature fusion mechanism; and insufficient consideration of real-time deployment constraints on edge nodes.

DETECX addresses these limitations through three modules: MTCE employs parallel dilated causal convolutions ($d \in \{1, 2, 4, 8\}$) to simultaneously capture short-term burst anomalies and long-term behavioural trends; CAFM establishes a cross-domain attention mechanism for dynamic injection of cloud-side global attack semantics into edge-side local detection; and EDM employs three diversity-promoting subnetworks with adaptive weighted voting to improve recall on low-frequency attack classes.

Comprehensive experiments on CICIDS2017, CICIOT2023, and UNSW-NB15 demonstrated average accuracy 99.18%, macro F1 97.38%, and AUC 99.56%, outperforming seven representative baselines including ET-BERT and FlowTransformer. Ablation studies verified independent module contributions (MTCE: 2.53 pp, CAFM: 2.03 pp, EDM: 1.37 pp, Cloud Layer: 1.70 pp Δ F1). Cloud-edge deployment achieves 5.8 ms edge latency with $300\times$ bandwidth reduction. Cross-dataset generalisation yields average cross-domain F1 of 0.8794, and all improvements are statistically significant ($p < 0.05$, Wilcoxon signed-rank test).

Future research will: (i) evaluate DETECX on TLS 1.3-dominant traffic captures; (ii) integrate federated learning for privacy-preserving collaborative training; (iii) incorporate adversarial robustness mechanisms against traffic obfuscation; and (iv) develop continual learning strategies to maintain detection efficacy under evolving attack patterns. DETECX establishes a strong foundation for next-generation IDS systems that simultaneously achieve high detection accuracy, low latency, and practical cloud-edge deployability.

Declarations

Author Contributions: Conceptualization, [A1] and [A2]; Methodology, [A1]; Software, [A1] and [A3]; Validation, [A1], [A2] and [A3]; Formal analysis, [A1]; Writing—original draft, [A1]; Writing—review and editing, [A2] and [A4]; Supervision, [A4]. 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 three benchmark datasets used in this study are publicly available. CICIDS2017 is accessible at <https://www.unb.ca/cic/datasets/ids-2017.html>; CICIOT2023 at <https://www.unb.ca/cic/datasets/iotdataset-2023.html>; and UNSW-NB15 at <https://research.unsw.edu.au/projects/unswnb15-dataset>. No new datasets were created. The source code is available from the corresponding author upon reasonable request.

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

References

- [1] Abusitta, A., Bellaïche, M., Dagenais, M. R., & Halabi, T. (2019). A deep learning approach for proactive multi-cloud cooperative intrusion detection system. *Future Generation Computer Systems*, 98, 308–318.

- [2] Aceto, G., Ciuonzo, D., Montieri, A., & Pescapé, A. (2019). MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. *Computer Networks*, 165, 106944.
- [3] Aceto, G., Ciuonzo, D., Montieri, A., & Pescapé, A. (2019). Mobile encrypted traffic classification using deep learning. *IEEE Transactions on Network and Service Management*, 16(2), 445–458.
- [4] Anderson, B., & McGrew, D. (2017). Machine learning for encrypted malware traffic classification. *Proceedings of the 23rd ACM SIGKDD*, 1723–1732.
- [5] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*.
- [6] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- [7] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176.
- [8] Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8), 1655–1674.
- [9] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD*, 785–794.
- [10] Cui, Y., Jia, M., Lin, T.-Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. *Proceedings of CVPR*, 9268–9277.
- [11] Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems*, 1–15. Springer.
- [12] Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection using deep learning for IoT. *Future Generation Computer Systems*, 82, 761–768.
- [13] Ferrag, M. A., Maglaras, L., Moschogiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection. *Journal of Information Security and Applications*, 50, 102419.
- [14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- [15] Khraisat, H., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems. *Cybersecurity*, 2, Article 20.
- [16] Lin, X., Xiong, G., Gou, G., Li, Z., Shi, J., & Yu, J. (2022). ET-BERT: A contextualized datagram representation with pre-training transformers. *Proceedings of the ACM Web Conference 2022*, 633–642.
- [17] Lotfollahi, M., Jafari Siavoshani, M., & Shirali Hossein Zade, R. (2020). Deep packet: Encrypted traffic classification using deep learning. *Soft Computing*, 24(3), 1999–2012.
- [18] Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M., & Portmann, M. (2024). FlowTransformer: A transformer framework for flow-based NIDS. *Expert Systems with Applications*, 241, 122564.

- [19] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning from decentralized data. *Proceedings of AISTATS*, 1273–1282.
- [20] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection. 2015 Military Communications and Information Systems Conference, 1–6.
- [21] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIOT2023: A real-time dataset for large-scale IoT attacks. *Sensors*, 23(13), 5941.
- [22] Rezaei, S., & Liu, X. (2019). Deep learning for encrypted traffic classification: An overview. *IEEE Communications Magazine*, 57(5), 76–81.
- [23] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & Security*, 86, 147–167.
- [24] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4), e1249.
- [25] Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39.
- [26] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset. *Proceedings of ICISSP*, 108–116.
- [27] Sharma, A., & Lashkari, A. H. (2025). A survey on encrypted network traffic. *Computer Networks*, 257, 110984.
- [28] Shen, M., Ye, K., Liu, X., Zhu, L., Kang, J., Yu, S., Li, Q., & Xu, K. (2023). Machine learning-powered encrypted network traffic analysis. *IEEE Communications Surveys & Tutorials*, 25(1), 791–824.
- [29] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646.
- [30] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for NIDS. 2010 IEEE Symposium on Security and Privacy, 305–316.
- [31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in NeurIPS*, 30, 5998–6008.
- [32] Velan, P., Čermák, M., Čeleda, P., & Drašar, M. (2015). A survey of methods for encrypted traffic classification. *International Journal of Network Management*, 25(5), 355–374.
- [33] Wang, W., Zhu, M., Wang, J., Zeng, X., & Yang, Z. (2017). End-to-end encrypted traffic classification with 1D CNNs. 2017 IEEE ISI, 43–48.
- [34] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM TIST*, 10(2), 1–19.
- [35] Zhou, S., Zhang, W., Liu, J., Guo, J., Zhou, H., Niu, W., Zhao, J., & Shen, X. S. (2019). Edge intelligence: Paving the last mile of AI with edge computing. *Proceedings of the IEEE*, 107(8), 1738–1762.