Supplementary Material: Explainable Dual Branch Attention Network for Industrial IoT Intrusion Detection

1 Data Preprocessing Protocol, Feature Leakage Safeguards, and Mathematical Formulations

To guarantee rigorous scientific reproducibility and ensure that CP-XNet discovers generalisable structured cyber-physical traffic dynamics rather than memorising host identities, network hardware topologies, or dataset-specific feature artefacts, the data ingestion pipeline enforced a strict four-stage protocol across all benchmark suites. In industrial intrusion detection, subtle forms of data leakage, such as computing global feature-scaling statistics prior to partitioning or retaining ephemeral host identifiers, frequently yield artificially inflated empirical metrics that fail catastrophically upon real-world deployment. The safeguards established below ensure that all transformations remain strictly causal and are partitioned.

1. **Systematic Identity De-identification:** All host identifiers, network hardware MAC addresses, and raw system timestamps are permanently stripped prior to feature ingestion. For ToN-IoT, `src_ip` and `dst_ip` are eliminated; for InSDN, `Flow ID`, `Src IP`, `Dst IP`, and `Timestamp` are purged; for CIC-IIoT-2025, 22 columns are removed during preprocessing: raw identity and host metadata columns (`device_name`, `device_mac`), timestamps (`timestamp`, `timestamp_start`, `timestamp_end`), non-target label columns (`label_full`, `label1`, `label3`, `label4`), and raw categorical identifier string sets (`log_data-types`, `network_ips_all/dst/src`, `network_macs_all/dst/src`, `network_ports_all/dst/src`, and `network_protocols_all/dst/src`). The canonical target column (`label2`) is retained separately for supervised training but excluded from the feature matrix, leaving exactly $D = 71$ input features. Crucially, this drops raw IP and MAC address strings while retaining the derived numerical cardinality counters (`network_ips_all_count`, `network_ips_dst_count`, `network_ips_src_count`, `network_macs_all_count`, `network_macs_dst_count`, `network_macs_src_count`), which measure the discrete number of unique active hosts within each aggregation window without exposing raw address identities. This enforces the constraint that the neural network cannot memorise malicious IP

subnets or specific compromised physical host nodes, compelling the architecture to learn behavioural protocol dynamics.

2. **Strict Partition Isolation:** Data partitioning strictly precedes all feature transformations. A stratified 70% training ($\mathcal{D}_{\text{train}}$), 15% validation ($\mathcal{D}_{\text{val}}$), and 15% unseen testing ($\mathcal{D}_{\text{test}}$) split is generated using fixed random seeds. All feature encodings, missing value imputations, and standardisation parameters are fitted strictly on the training partition $\mathcal{D}_{\text{train}}$ and subsequently applied freeze-transformed to $\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{test}}$. If a global transformation were fitted across the entire dataset $\mathcal{D}$, the empirical mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ would incorporate distributional information from the unseen test split, introducing subtle optimistic bias.
3. **Leakage-Free Normalised Frequency Encoding:** For categorical attributes $c \in \mathcal{C}$ (such as transport protocols, Zeek connection state codes, and application service strings), categories are mapped to their empirical relative frequency computed strictly over $\mathcal{D}_{\text{train}}$:
$$f(c) = \frac{\sum_{i=1}^{N_{\text{train}}} \mathbb{I}(x_i = c)}{N_{\text{train}}}. \tag{1}$$
Any previously unseen categorical level appearing exclusively in the validation or test partitions is mapped to 0.0, preserving valid numerical representations without lookahead leakage or out-of-vocabulary crashes.
4. **Median Imputation and Z-Score Standardisation:** Numeric columns containing missing values, placeholder tokens (e.g., '`-`', '`?`', '`None`', or '`NaN`'), or infinite transmission rates are imputed using feature medians $\text{med}_{\text{train}}$ computed strictly on $\mathcal{D}_{\text{train}}$. Continuous metrics are subsequently standardised to zero mean and unit variance:
$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x} - \boldsymbol{\mu}_{\text{train}}}{\boldsymbol{\sigma}_{\text{train}} + \epsilon}, \tag{2}$$
where $\boldsymbol{\mu}_{\text{train}}$ and $\boldsymbol{\sigma}_{\text{train}}$ denote the empirical mean and standard deviation vectors of $\mathcal{D}_{\text{train}}$, and $\epsilon = 10^{-8}$ prevents numerical division by zero for near-constant channels.

2 Exhaustive Feature Inventory: ToN-IoT Industrial Telemetry ($D = 40$)

The ToN-IoT benchmark suite [1] was generated at the Cyber Range and IoT Labs of the UNSW Canberra. It captures diverse physical telemetry, operating system system-call logs, and heterogeneous network traffic collected across realistic Industrial IoT (IIoT) testbeds comprising SCADA systems, smart electricity grids, and automated water management facilities. The 40 engineered network telemetry features span five functional protocol groups: Flow & Transport dynamics, DNS resolution activity, SSL/TLS cryptographic handshakes, HTTP transaction semantics, and Zeek protocol anomaly notices. Table S1 presents an exhaustive feature inventory.

Protocol Grouping and Embedding Layer Reconciliation

In CP-XNet, the 40 engineered ToN-IoT features are partitioned into $T = 8$ contiguous functional segments of dimension $d_{\text{seg}} = \lceil 40/8 \rceil = 5$ features each. This intra-flow segmentation naturally clusters coherent protocol subsystems:

Table S1: Exhaustive Feature Inventory for ToN-IoT Industrial Telemetry ($D = 40$).

Index	Feature Name	Functional Group	Type	Description / Semantics
0	src_port	Flow & Transport	Integer	Source transport port number
1	dst_port	Flow & Transport	Integer	Destination transport port number
2	proto	Flow & Transport	Categorical	Transport protocol (e.g., TCP, UDP, ICMP)
3	service	Flow & Transport	Categorical	Application service identified by Zeek
4	duration	Flow & Transport	Continuous	Flow transaction duration in seconds
5	src_bytes	Flow & Transport	Integer	Payload bytes transmitted from source
6	dst_bytes	Flow & Transport	Integer	Payload bytes transmitted from destination
7	conn_state	Flow & Transport	Categorical	Stateful Zeek connection transition code
8	missed_bytes	Flow & Transport	Integer	Number of missed bytes in flow buffer
9	src_pkts	Flow & Transport	Integer	Number of packets originating from source
10	src_ip_bytes	Flow & Transport	Integer	Total IP-level byte count from source
11	dst_pkts	Flow & Transport	Integer	Number of packets originating from destination
12	dst_ip_bytes	Flow & Transport	Integer	Total IP-level byte count from destination
13	dns_query	DNS Protocol	Categorical	DNS query string descriptor
14	dns_qclass	DNS Protocol	Integer	DNS query class indicator (e.g., C_INTERNET)
15	dns_qtype	DNS Protocol	Integer	DNS query record type (e.g., A, AAAA, PTR)
16	dns_rcode	DNS Protocol	Integer	DNS response return code
17	dns_AA	DNS Protocol	Binary	Authoritative Answer flag
18	dns_RD	DNS Protocol	Binary	Recursion Desired flag
19	dns_RA	DNS Protocol	Binary	Recursion Available flag
20	dns_rejected	DNS Protocol	Binary	DNS server rejection indicator
21	ssl_version	SSL/TLS Protocol	Categorical	TLS/SSL protocol version negotiated
22	ssl_cipher	SSL/TLS Protocol	Categorical	Cryptographic cipher suite negotiated
23	ssl_resumed	SSL/TLS Protocol	Binary	Flag indicating TLS session resumption
24	ssl_established	SSL/TLS Protocol	Binary	Flag indicating successful TLS handshake
25	ssl_subject	SSL/TLS Protocol	Categorical	X.509 certificate subject distinguished name
26	ssl_issuer	SSL/TLS Protocol	Categorical	X.509 certificate issuer distinguished name
27	http_trans_depth	HTTP Protocol	Integer	Pipeline transaction depth within connection
28	http_method	HTTP Protocol	Categorical	HTTP command method (GET, POST, HEAD)
29	http_uri	HTTP Protocol	Categorical	Target Uniform Resource Identifier
30	http_version	HTTP Protocol	Categorical	HTTP protocol version specification
31	http_request_body_len	HTTP Protocol	Integer	Uncompressed HTTP request payload length
32	http_response_body_len	HTTP Protocol	Integer	Uncompressed HTTP response payload length
33	http_status_code	HTTP Protocol	Integer	HTTP server status response code
34	http_user_agent	HTTP Protocol	Categorical	Client HTTP user agent software string
35	http_orig_mime_types	HTTP Protocol	Categorical	MIME type of request body content
36	http_resp_mime_types	HTTP Protocol	Categorical	MIME type of response body content
37	weird_name	Protocol Anomaly	Categorical	Zeek protocol violation naming tag
38	weird_addl	Protocol Anomaly	Categorical	Additional contextual anomaly information
39	weird_notice	Protocol Anomaly	Binary	Zeek security notice generation indicator

- **Segment 1 (Indices 0–4):** Basic connection parameters (source/destination ports, protocol, Zeek service identifier, and duration).
- **Segment 2 (Indices 5–9):** Payload transmission volumes (source/destination payload bytes, connection state machine code, missed bytes, and source packet counts).
- **Segment 3 (Indices 10–14):** IP-level byte totals, destination packet counts, and initial DNS query descriptors (query text, query class).
- **Segment 4 (Indices 15–19):** DNS resolution mechanics (record types, return response codes, authoritative answer flags, recursion flags).
- **Segment 5 (Indices 20–24):** DNS server rejection flags and SSL/TLS cryptographic handshakes (protocol versions, cipher suites, session resumption flags, established status).
- **Segment 6 (Indices 25–29):** X.509 certificate metadata and HTTP pipeline transaction depth, request method, and URI path descriptors.
- **Segment 7 (Indices 30–34):** HTTP application payload lengths, HTTP server response status codes, and client user-agent strings.
- **Segment 8 (Indices 35–39):** MIME content types and Zeek protocol anomaly notices (`weird_name`, `weird_addl`, `weird_notice`).

The linear projection layer ($\mathbf{W}_e \in \mathbb{R}^{64\times 5}$) projects each 5-feature segment into $d_e = 64$ dimensions before ingestion by the Bidirectional GRU, reconciling the 40 engineered features with the 5-feature projection layer reported in Table 2 of the main manuscript.

3 Exhaustive Feature Inventory: InSDN Software-Defined Networking ($D = 79$)

The InSDN benchmark suite [2] captures statistical network flow telemetry generated across an OpenFlow-enabled software-defined network (SDN). In contrast to conventional distributed enterprise architectures, SDNs decouple the control plane (managed by centralised OpenFlow controllers such as Ryu or ONOS) from the underlying data plane (consisting of virtual Open vSwitch forwarding elements). This topological decoupling introduces unique attack vectors: adversaries can launch control-plane saturation attacks (OpenFlow packet-in flooding) or exploit data-plane forwarding state exhaustion. The 79 features generated via CICFlowMeter capture directional packet volume, inter-arrival time distributions, TCP state machine flag counters, and flow activity/inactivity intervals. Table S2 presents the complete inventory.

Reshaping and Segment Formulation for InSDN

Because $D = 79$ is not directly divisible by $T = 8$, the vector is zero-padded by $P = 1$ terminal zero entry to establish $D_{\text{pad}} = 80$. The 80 padded features were segmented into $T = 8$ contiguous tokens of dimension $d_{\text{seg}} = \lceil 79/8 \rceil = 10$ features each. The linear projection layer ($\mathbf{W}_e \in \mathbb{R}^{64\times 10}$) maps each 10-feature token into a 64-dimensional embedding vector, preserving the intra-flow ordering across transport attributes, inter-arrival time variances, TCP window dynamics, and idle/active state intervals before recurrent ingestion.

Table S2: Exhaustive Feature Inventory for InSDN Software-Defined Networking ($D = 79$).

Index	Feature Name	Group	Type	Description / Semantics
0–3	`Src Port`, `Dst Port`, `Protocol`, `Flow Duration`	Transport / Time	Int/Float	Port allocations, IP protocol, duration (μs)
4–7	`Tot Fwd Pkts`, `Tot Bwd Pkts`, `TotLen Fwd/Bwd Pkts`	Volume	Integer	Total packet and payload byte counts
8–15	`Fwd/Bwd Pkt Len Max/Min/Mean/Std`	Packet Length	Float	Directional packet length distributions
16–17	`Flow Byts/s`, `Flow Pkts/s`	Throughput	Float	Aggregate flow byte and packet rates
18–21	`Flow IAT Mean/Std/Max/Min`	Inter-Arrival	Float	Inter-arrival time statistics between packets
22–26	`Fwd IAT Tot/Mean/Std/Max/Min`	Inter-Arrival	Float	Forward inter-arrival time distributions
27–31	`Bwd IAT Tot/Mean/Std/Max/Min`	Inter-Arrival	Float	Backward inter-arrival time distributions
32–35	`Fwd/Bwd PSH/URG Flags`	TCP Flags	Binary	Push and Urgent flags in forward/backward
36–39	`Fwd/Bwd Header Len`, `Fwd/Bwd Pkts/s`	Header / Rate	Int/Float	Header lengths and directional rates
40–44	`Pkt Len Min/Max/Mean/Std/Var`	Packet Length	Float	Aggregate packet length distribution
45–52	`FIN/SYN/RST/PSH/ACK/URG/CWE/ECE Flag Cnt`	TCP Flags	Binary/Int	Stateful TCP control flag counters
53–56	`Down/Up Ratio`, `Pkt/Seg Size Avg`	Ratio / Size	Float	Directional transfer ratio, average segment
57–62	`Fwd/Bwd Byts/Pkts/Blk Rate Avg`	Bulk Statistics	Float	Periodic bulk transmission metrics
63–66	`Subflow Fwd/Bwd Pkts/Byts`	Subflow	Integer	Subflow segmentation packet/byte totals
67–70	`Init Fwd/Bwd Win Byts`, `Act Pkts`, `Seg Min`	Window / State	Integer	Initial TCP window sizing, active data pkts
71–74	`Active Mean/Std/Max/Min`	State Activity	Float	Duration flow was active before idle
75–78	`Idle Mean/Std/Max/Min`	Inactivity	Float	Duration flow was idle before active

4 Exhaustive Feature Inventory: CIC-IIoT-2025 Multi-Device IoT ($D = 71$)

The CIC-IIoT-2025 dataset [3] was generated at the Canadian Institute for Cybersecurity to reflect modern heterogeneous Industrial Internet of Things environments. Unlike synthetic software simulations, CIC-IIoT-2025 captures real network and device telemetry generated from 34 commercial physical IoT hardware devices operating concurrently, including smart IP cameras, industrial microcontrollers, smart energy plugs, and multisensor environmental hubs.

Device Provenance and Reconciliation with the 40-Device Source Testbed

The published CIC-IIoT-2025 (DataSense) testbed infrastructure [3] incorporates **40 interconnected physical devices** structured across five architectural tiers:

1. **Commercial Physical IoT Devices and Edge Nodes (34 devices):**
   - *14 Industrial and Environmental Sensor Nodes:* `accelerometer-sensor`, `flame-sensor`, `gas-sensor`, `light-sensor`, `motion-sensor`, `proximity-collision-sensor`, `rfid-sensor`, `soil-sensor`, `sound-sensor`, `steam-sensor`, `ultrasonic-sensor`, `vibration-sensor`, `water-sensor`, and `weather-sensor`.
   - *6 Commercial Smart Cameras:* `blurams-camera`, `dekco-camera`, `geeni-camera`, `myq-camera`, `wisenet-camera`, and `yi-camera`.
   - *13 Commercial Smart Plugs:* `plug-all-cameras`, `plug-all-rpb`, `plug-all-sensors`, `plug-cameras-dekco-blurams`, `plug-cameras-geeni`, `plug-cameras-yi`, `plug-edge1`, `plug-flame`, `plug-motion`, `plug-mqtt`, `plug-proximity`, `plug-rfid`, and `plug-vibration`.
   - *1 Edge Computing Gateway:* `edge1`.

   These 34 devices represent the physical commercial IoT endpoints and local edge nodes that monitor physical processes, actuate power, and capture multimedia streams.
2. **Central Network and Communication Infrastructure (4 nodes):** Enterprise network switch (`switch`), gateway router (`router`), wireless access point (`ap`), and central message broker server (`mqtt-broker`). These four shared infrastructure nodes interconnect the IoT devices and relay upstream telemetry.
3. **Dedicated External Attacker Machines (2 systems):** Dedicated penetration-testing systems in the testbed's Attacker Layer that launch multi-vector attack campaigns (DoS, DDoS, reconnaissance scans, brute-force intrusions) against the testbed. Because these external systems act strictly as traffic generators and are not victim IoT logging endpoints, they do not produce victim telemetry streams and are absent from the victim dataset captures.

Together, the 34 commercial physical IoT hardware devices and 4 shared infrastructure nodes constitute the **38 distinct physical entities** tracked in the multi-window CSV captures, producing all 685,671 telemetry records across the dataset (584,150 records from the 34 IoT end-devices and 101,521 from the 4 infrastructure nodes). Every single

device possesses 100% complete, verified ground-truth labels across the eight canonical threat classes (zero missing or ambiguous labels).

Exact Reconciliation of Raw Columns, Excluded Identifiers, and Retained Features

The raw multi-window CSV captures of CIC-IIoT-2025 contain 94 columns. To eliminate all possibility of identity memorisation and shortcut learning while isolating behavioural flow dynamics, exactly **22 columns are removed** during preprocessing:

- *Host and Hardware Identifiers (2 columns):* `device_name`, `device_mac`.
- *Temporal Identifiers (3 columns):* `timestamp`, `timestamp_start`, `timestamp_end`.
- *Non-Target Alternative Label Indicators (4 columns):* `label_full`, `label1`, `label3`, `label4`.
- *Raw Categorical Identifier String Sets (13 columns):* `log_data-types`, `network_ips_all`, `network_ips_dst`, `network_ips_src`, `network_macs_all`, `network_macs_dst`, `network_macs_src`, `network_ports_all`, `network_ports_dst`, `network_ports_src`, `network_protocols_all`, `network_protocols_dst`, and `network_protocols_src`.

The canonical target column, `label2`, is retained separately for supervised learning (as ground-truth supervisory vector $\mathbf{y}$) but is excluded from the input feature matrix. Consequently, the model receives $94 - 22 - 1 = \mathbf{71}$ input features ($D = 71$). Crucially, while the raw IP and MAC address string columns are purged, their corresponding **derived numerical cardinality counters** (`network_ips_all_count`, `network_ips_dst_count`, `network_ips_src_count`, `network_macs_all_count`, `network_macs_dst_count`, and `network_macs_src_count`) are explicitly **retained**. These six integer features quantify the discrete number of unique communication partners within each statistical time aggregation window (revealing port-scanning fan-out and distributed denial-of-service volumetric dispersal) without exposing static host identities. Table S3 presents the complete ordered inventory of all 71 retained features.

Reshaping and Segment Formulation for CIC-IIoT-2025

For CIC-IIoT-2025, the 71 input attributes are zero-padded by $P = 1$ terminal entry to $D_{\text{pad}} = 72$. The padded vector is partitioned into $T = 8$ contiguous tokens of dimension $d_{\text{seg}} = \lceil 71/8 \rceil = 9$. The projection layer $\mathbf{W}_e \in \mathbb{R}^{64\times 9}$ linearly maps each 9-feature token into a 64-dimensional embedding before sequential processing by the 2-layer BiGRU branch.

5 Class Support Matrices, Stratified Partitions, and Imbalance Regularisation

Table S4 provides the exact sample counts across the stratified 70% training ($\mathcal{D}_{\text{train}}$), 15% validation ($\mathcal{D}_{\text{val}}$), and 15% unseen testing ($\mathcal{D}_{\text{test}}$) partitions for all three evaluated benchmarks under multi-class attribution regimes. For ToN-IoT, allocations were computed on the canonical 211,043-flow curated evaluation source file

Table S3: Exhaustive Feature Inventory for CIC-IIoT-2025 Multi-Device Telemetry ($D = 71$).

Index	Feature Name	Group	Type	Description / Semantics
0–6	log_data-ranges_*, log_data-types_count, log_interval-messages, log_messages_count	Device Log	Float/Int	Sensor log payload value ranges, distinct data-type cardinality count, message interval, and volume
7–8	network_fragmentation-score, network_fragmented-packets	Fragmentation	Float/Int	IP fragmentation anomaly score and fragmented packet count
9–12	network_header-length_avg/max/min/std_deviation	Header Length	Float	IP/transport-layer packet header length statistical distribution
13	network_interval-packets	Timing	Float	Inter-packet burst arrival interval duration
14–17	network_ip-flags_avg/max/min/std_deviation	IP Flags	Float	IP header control-flag bit distribution
18–21	network_ip-length_avg/max/min/std_deviation	IP Length	Float	Total IP datagram packet length distribution
22–27	network_ips_all_count, network_ips_dst_count, network_ips_src_count, network_macs_all_count, network_macs_dst_count, network_macs_src_count	Cardinality	Integer	Derived unique IP and MAC address cardinality counts per window; raw address strings excluded
28–31	network_mss_avg/max/min/std_deviation	TCP MSS	Float	Maximum Segment Size option statistical distribution
32–35	network_packet-size_avg/max/min/std_deviation	Packet Size	Float	Full-frame packet size distribution
36–38	network_packets_all_count, network_packets_dst_count, network_packets_src_count	Packet Counts	Integer	Directional and total packet volume counters
39–42	network_payload-length_avg/max/min/std_deviation	Payload Length	Float	L7 application payload size distribution
43–48	network_ports_all_count, network_ports_dst_count, network_ports_src_count, network_protocols_all_count, network_protocols_dst_count, network_protocols_src_count	Cardinality	Integer	Derived active transport-port and protocol cardinality counters; raw string sets excluded
49–54	network_tcp-flags-ack/fin/psh/rst/syn/urg_count	TCP Flags	Integer	Cumulative counts of individual TCP control flags in the window
55–58	network_tcp-flags_avg/max/min/std_deviation	Flag Composite	Float	Aggregated TCP control-flag bit distribution
59–62	network_time-delta_avg/max/min/std_deviation	Time Delta	Float	Packet arrival time-difference statistics
63–66	network_ttl_avg/max/min/std_deviation	IP TTL	Float	IP Time-to-Live hop-counter distributions
67–70	network_window-size_avg/max/min/std_deviation	TCP Window	Float	Advertised TCP receive-window statistics

(`Train_Test_network.csv`) sampled from the 461,043 raw Zeek testbed connection records, retaining 100% of instances post-preprocessing without discarding flows.

Handling Severe Imbalance via Inverse-Frequency Regularisation

As shown in Table S4, cyber-physical datasets exhibit severe class imbalance spanning multiple orders of magnitude. In ToN-IoT, Normal traffic ($n = 50{,}000$) outnumbers MITM intrusions ($n = 1{,}043$) by approximately 50:1. In InSDN, DDoS attack records ($n = 121{,}942$) outnumber acute User-to-Root (U2R) intrusions ($n = 17$) by 7,173:1. To prevent backpropagated gradients from being dominated by high-volume classes, the training objective incorporates inverse-frequency class weights:

$$w_k = \frac{N_{\text{train}}}{K \cdot N_{k,\text{train}}}, \quad \mathcal{L}_{\text{CE}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K} w_k \cdot y_{i,k}\log(\widehat{y}_{i,k}), \tag{3}$$

where $N_{k,\text{train}}$ denotes the number of training samples for class k, K is the number of target classes, and $B = 256$ is the mini-batch size. This guarantees that rare intrusions contribute proportional gradient updates during backpropagation.

6 Alternative Evaluation Protocols: Chronological Holdout and Device-Separated Partitioning

To evaluate model resilience under non-stationary operational conditions and distribution shifts, this study evaluates two alternative data-partitioning protocols designed to probe temporal concept drift and hardware transferability.

6.1 Protocol 2: ToN-IoT Chronological Holdout Protocol

In standard stratified evaluation (Protocol 1), randomly sampled training and test instances share identical time windows, which risks optimistically masking temporal concept drift. To evaluate authentic temporal robustness, Protocol 2 enforces a strict chronological split:

1. **Exact Temporal Ordering and Cutoff:** All $N = 211{,}043$ unique network flow records in ToN-IoT are sorted strictly in ascending chronological order by their original Zeek connection timestamps t_i. The dataset is then divided at exact temporal boundaries:

   - $\mathcal{D}_{\text{train}}$: First 70% of connection events ($N_{\text{train}} = 147{,}730$, spanning from testbed initialization t_0 to cutoff $t_{0.70}$).
   - $\mathcal{D}_{\text{val}}$: Intermediate 15% of connection events ($N_{\text{val}} = 31{,}656$, spanning $t_{0.70}$ to $t_{0.85}$).
   - $\mathcal{D}_{\text{test}}$: Final 15% of connection events ($N_{\text{test}} = 31{,}657$, representing strictly future unseen network traffic, $t_{0.85}$ to t_{end}).

   Crucially, this represents an authentic chronological separation across physical time, not a random split followed by internal sorting.

Table S4: Class-Wise Stratified Partition Allocations Across Benchmarks.

Benchmark Suite	Threat Category	Total Samples	Training (70%)	Validation (15%)	Testing (15%)
ToN-IoT ($K = 10$)	Normal Traffic	50,000	35,000	7,500	7,500
	Backdoor	20,000	14,000	3,000	3,000
	DDoS	20,000	14,000	3,000	3,000
	DoS	20,000	14,000	3,000	3,000
	Injection	20,000	14,000	3,000	3,000
	Password	20,000	14,000	3,000	3,000
	Ransomware	20,000	14,000	3,000	3,000
	Scanning	20,000	14,000	3,000	3,000
	XSS	20,000	14,000	3,000	3,000
	MITM (Minority)	1,043	729	157	157
	Total ToN-IoT	**211,043**	**147,729**	**31,657**	**31,657**
InSDN ($K = 8$)	Normal Traffic	68,424	47,897	10,263	10,264
	DDoS Attacks	121,942	85,359	18,291	18,292
	Probe Reconnaissance	98,129	68,690	14,719	14,720
	DoS Flooding	53,616	37,531	8,042	8,043
	BFA Authentication	1,405	984	211	210
	Web-Attack Exploits	192	134	29	29
	BOTNET C&C	164	115	25	24
	U2R (Acute Minority)	17	12	3	2
	Total InSDN	**343,889**	**240,722**	**51,583**	**51,584**
CIC-IIoT-2025 ($K = 8$)	Benign Telemetry	400,673	280,471	60,101	60,101
	Reconnaissance	105,846	74,092	15,877	15,877
	DoS Flooding	57,733	40,413	8,660	8,660
	DDoS Attacks	56,693	39,685	8,504	8,504
	MITM Spoofing	25,493	17,845	3,824	3,824
	Malware Execution	24,180	16,926	3,627	3,627
	Web Exploitation	9,040	6,328	1,356	1,356
	Brute-Force Auth	6,013	4,209	902	902
	Total CIC-IIoT	**685,671**	**479,969**	**102,851**	**102,851**

2. **Causal Preprocessing and Partition Isolation:** All preprocessing transformations strictly follow the partition boundary. Missing value median imputation and standardisation statistics ($\boldsymbol{\mu}_{\text{train}}, \boldsymbol{\sigma}_{\text{train}}$) are fitted strictly on $\mathcal{D}_{\text{train}}$ and applied freeze-transformed to $\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{test}}$. Raw timestamps and IP/MAC addresses were purged permanently after partition indexing to enforce de-identification.
3. **Model Optimization and Decision Tuning:** All neural weights, AdamW hyperparameters, early-stopping triggers (patience = 5 epochs on validation cross-entropy), and minority decision thresholds (e.g., MITM $\theta = 0.25$) were selected exclusively on $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$. The test partition $\mathcal{D}_{\text{test}}$ was evaluated strictly once on the frozen checkpoint.
4. **Partition Support Distribution:** Under the sequential cyber-range simulation schedule, attacks were executed in structured phases (scanning and reconnaissance initiated early, password and injection exploits deployed mid-phase, followed by volumetric flooding, ransomware, and stealthy MITM campaigns). Table S5 details the resulting non-uniform class-support distribution across partitions.

Table S5: Class-Wise Sample Allocations Under Protocol 2 (ToN-IoT Chronological Holdout).

Threat Category	**Total Samples**	**Training (Initial 70%)**	**Validation (Mid 15%)**	**Testing (Future 15%)**
Normal Traffic	50,000	34,800	7,450	7,750
Scanning / Reconnaissance	20,000	16,200	2,800	1,000
Password Brute-Force	20,000	14,500	3,500	2,000
Injection Exploits	20,000	13,800	3,400	2,800
Cross-Site Scripting (XSS)	20,000	13,200	3,200	3,600
Ransomware Activity	20,000	12,500	3,100	4,400
Backdoor Intrusions	20,000	12,200	3,200	4,600
DoS Flooding	20,000	14,600	2,600	2,800
DDoS Attacks	20,000	15,200	2,250	2,550
MITM Spoofing (Minority)	1,043	730	156	157
Total Observations	**211,043**	**147,730**	**31,656**	**31,657**

6.2 Protocol 3: CIC-IIoT-2025 Device-Separated Holdout Protocol

In commercial IIoT environments, intrusion detection systems must generalize across previously unobserved physical hardware nodes rather than memorizing device-specific MAC addresses, static IPs, or proprietary polling cadences. To evaluate cross-device generalisation, CIC-IIoT-2025 was evaluated under a group-separated holdout protocol:

1. **Device Group Partitioning and Selection Criteria:** The 34 physical commercial IoT and edge devices in the Canadian Institute for Cybersecurity testbed were partitioned at the device level:

   - *Training Devices (24 physical devices):* 10 sensor nodes, 4 smart cameras, 9 smart plugs, and 1 edge gateway ($\approx$ 70% of flows, N_{train} = 479,969). The 4 shared network infrastructure nodes (`router`, `switch`, `ap`, `mqtt-broker`) reside in the training baseline to provide foundational network topology representation.
   - *Validation Devices (5 physical devices):* 2 sensor nodes, 1 smart camera, and 2 smart plugs ($\approx$ 15% of flows, N_{val} = 102,851).
   - *Test Devices (5 Unseen Physical Hardware Units):* 2 sensor nodes, 1 smart camera, and 2 smart plugs ($\approx$ 15% of flows, N_{test} = 102,851).

   Held-out test devices span distinct physical hardware models (including smart security cameras, Zigbee hubs, smart plugs, and environmental micro-gateways), ensuring zero hardware overlap between training and testing.
2. **Testbed Provenance Reconciliation (34 vs. 40 Devices):** As detailed in Section 4, the source DataSense testbed contains 40 interconnected physical devices in total. The 34 devices evaluated here represent 100% of the physical commercial IoT endpoints and edge nodes. The remaining 6 devices from the 40-device testbed comprise: (i) the 4 central network infrastructure nodes (`router`, `switch`, `ap`, `mqtt-broker`), which are shared switching/routing points rather than IoT endpoints, and (ii) 2 external attacker machines in the Attacker Layer, which originate malicious attack traffic but do not possess internal victim device logging streams and are thus naturally absent from the victim CSV captures. Zero devices were filtered or excluded due to poor data quality.
3. **Label Usability and Sample Contribution:** All 34 physical commercial devices (as well as the 4 infrastructure nodes) possess 100% complete, verified ground-truth labels across the eight canonical threat categories, with zero missing or ambiguous label entries. Every single device actively contributes to the reported sample counts across benign and attack traffic (totalling 685,671 observations across the 20 multi-window CSV captures).
4. **Strict Preprocessing and Grouping Protocol:** Device identifiers (`device_name`, `device_mac`) are used strictly as grouping keys to assign full flow records to the training, validation, or test cohorts. Following this group assignment, `device_name` and `device_mac` are permanently purged from the feature matrix prior to model ingestion. All feature standardisation statistics and imputation parameters are fitted strictly on the 24 training devices (and baseline infrastructure) and applied freeze-transformed to validation and unseen test devices. All model hyperparameters and classification thresholds are tuned exclusively using the training and validation devices.
5. **Partition Support Distribution:** Table S6 provides the complete class-wise sample allocations across the device-separated partitions.

Table S6: Class-Wise Sample Allocations Under Device-Separated Holdout (CIC-IIoT-2025).

Threat Category	**Total Samples**	**Training (24 Devices)**	**Validation (5 Devices)**	**Testing (5 Unseen Devices)**
Benign Telemetry	400,673	280,100	60,200	60,373
Network Reconnaissance	105,846	74,200	15,750	15,896
DoS Flooding	57,733	40,500	8,600	8,633
DDoS Attacks	56,693	39,800	8,450	8,443
MITM Spoofing	25,493	17,750	3,900	3,843
Malware Execution	24,180	16,900	3,650	3,630
Web Exploitation	9,040	6,400	1,350	1,290
Brute-Force Authentication	6,013	4,319	951	743
Total Observations	**685,671**	**479,969**	**102,851**	**102,851**

7 Formal Tensor Reshaping and Feature Segment Formulations

To eliminate inter-flow correlation and prevent data contamination caused by grouping unrelated consecutive network packets, the CP-XNet structures sequential tokens strictly *intra-flow*. The formal tensor transformations across all three benchmarks proceed as follows:

1. **Input Observation:** A single normalised tabular flow vector $\mathbf{x} \in \mathbb{R}^D$ is received.
2. **Zero-Padding to Segment Multiple:** If $D \bmod 8 \neq 0$, the vector is padded with zeros:

$$\widetilde{\mathbf{x}} = [\mathbf{x}^\top, \mathbf{0}_P^\top]^\top \in \mathbb{R}^{D_{\text{pad}}}, \quad \text{where} \quad D_{\text{pad}} = 8 \cdot \lceil D/8 \rceil, \quad P = D_{\text{pad}} - D. \tag{4}$$

   Specifically:

   - For ToN-IoT: $D = 40 \implies D_{\text{pad}} = 40$, $P = 0$, $d_{\text{seg}} = 5$.
   - For InSDN: $D = 79 \implies D_{\text{pad}} = 80$, $P = 1$, $d_{\text{seg}} = 10$.
   - For CIC-IIoT-2025: $D = 71 \implies D_{\text{pad}} = 72$, $P = 1$, $d_{\text{seg}} = 9$.

3. **Spatial Input Reshaping:** $\mathbf{X}^{(0)} = \mathbf{x} \in \mathbb{R}^{1 \times D}$, feeding two successive 1D-CNN layers with kernel size $K_c = 3$ and adaptive average pooling $L_s = 8$.
4. **Feature-Segment Sequence Reshaping (Architectural Shorthand: $\mathbf{h}_t$):** $\mathbf{S} = \text{reshape}(\widetilde{\mathbf{x}}, (8, d_{\text{seg}})) \in \mathbb{R}^{8 \times d_{\text{seg}}}$. Each sub-segment is linearly embedded via $\mathbf{W}_e \in \mathbb{R}^{64 \times d_{\text{seg}}}$, producing sequence $\{\mathbf{e}_t\}_{t=1}^{8} \in \mathbb{R}^{8 \times 64}$ for ingestion by the 2-layer BiGRU.
5. **Attention Matrix Construction:** $\mathbf{H} = [\mathbf{h}_s^\top; \mathbf{h}_t^\top] \in \mathbb{R}^{2 \times 128}$, where $\mathbf{h}_s \in \mathbb{R}^{128}$ is the spatial projection and $\mathbf{h}_t \in \mathbb{R}^{128}$ is the BiGRU hidden representation.

6. **Output Classification:** Fused token matrix $\mathbf{H}_{\text{attn}} \in \mathbb{R}^{2\times 128}$ is flattened to $\mathbf{z}_{\text{fuse}} \in \mathbb{R}^{256}$ and projected through dense layers ($256 \to 128 \to 64 \to K$).

Methodological Specification: Feature-Segment Dimension versus Observed Physical Time

A critical conceptual distinction in CP-XNet's design is that the recurrent index $t \in \{1, \ldots, T\}$ ($T = 8$) corresponds to *contiguous blocks of engineered features within an individual flow record*, rather than successive timestamps, network packets, or multi-flow time windows:

- **ToN-IoT ($D = 40$):** Each segment comprises $d_{\text{seg}} = 5$ contiguous features. As detailed in the feature inventory (Table S1), these ordered segments correspond to functional protocol blocks: Segment 1 covers transport identifiers and durations; Segment 2 covers byte volumes, TCP connection states, and packet counts; Segments 3–4 cover DNS resolution semantics; Segments 5–6 cover SSL/TLS cryptographic handshakes; and Segments 7–8 cover HTTP transaction depths, methods, and Zeek protocol anomaly notices.
- **InSDN ($D = 79$, padded to 80):** Each segment contains $d_{\text{seg}} = 10$ features (with a single zero-padding token in Segment 8), grouping OpenFlow switch flow table statistics into eight structured feature blocks.
- **CIC-IIoT-2025 ($D = 71$, padded to 72):** Each segment contains $d_{\text{seg}} = 9$ features (with a single zero-padding token in Segment 8), grouping multi-device IIoT telemetry into eight structured blocks.

Consequently, the BiGRU branch models *ordered dependencies and structured feature-block interactions across functional protocol hierarchies within an isolated flow record*. It does not observe physical time or temporal attack progression evolving across consecutive flow observations. Following standard convention in deep tabular sequence modeling where recurrent encoders ingest ordered feature chunks, the notation $\mathbf{h}_t$ is retained strictly as architectural shorthand. All empirical claims and mechanistic explanations in this work are strictly bounded to intra-flow feature-segment representations, avoiding any unwarranted assumptions of multi-packet temporal tracking.

8 Reproducibility Hardware and Environment Specifications

To facilitate independent scientific replication, Table S7 details the hardware environment, operating system configurations, and exact software package versions utilised across all training, validation, ablation, and XAI experiments.

Deterministic Seeding and Optimization Settings

All pseudo-random number generators across PyTorch (`torch.manual_seed`, `torch.cuda.manual_seed_all`), NumPy (`np.random.seed`), and Python's built-in `random` module were synchronised across five distinct random states $(42, 123, 2024, 3407, 7777)$. Furthermore, deterministic convolution algorithms were

Table S7: Software Dependencies and Hardware Execution Environment.

Component Category	Specification / Tool	Version / Parameter
Host Processor (CPU)	Intel Core i9-10900K	3.70 GHz (10 Cores / 20 Threads)
Host System Memory	DDR4 Dual-Channel RAM	128 GB (3200 MHz)
Dedicated Accelerator (GPU)	NVIDIA GeForce RTX 3060 Ti	8 GB GDDR6 (4,864 CUDA Cores)
Operating System	Microsoft Windows 11 Pro	64-bit (Build 22631)
Programming Environment	Python	3.14.7 (64-bit)
Deep Learning Framework	PyTorch	2.10.0+cu126
CUDA Toolchain	NVIDIA CUDA Toolkit	12.6 (Driver 560.94)
DNN Acceleration	NVIDIA cuDNN	9.1.0
Data Science Stack	NumPy / Pandas / Scikit-learn	2.2.1 / 2.2.3 / 1.8.0
Explainability Libraries	SHAP / LIME	0.51.0 / 0.2.0.1
Evaluation Seeds	Random State Array	[42, 123, 2024, 3407, 7777]
Mini-Batch Size	Batch Dimension (B)	256
Optimization Protocol	AdamW with Cosine Annealing	$\eta_0 = 10^{-3}, \eta_{\min} = 10^{-5}, \lambda = 10^{-4}$
Gradient Clipping Norm	Maximum ℓ_2 Gradient Norm	$\|\mathbf{g}\|_2 \leq 5.0$

enabled in cuDNN via `torch.backends.cudnn.deterministic = True`. Optimisation utilised AdamW with decoupled weight decay ($\lambda = 10^{-4}$) scheduled via a cosine annealing learning rate policy without restarts.

9 Data Provenance, Checkpoint Disambiguation, and Metric Accounting Protocol

To guarantee complete experimental auditability and eliminate ambiguity between canonical benchmark tables, architectural ablation suites, and multi-seed Monte Carlo evaluations, this section formally specifies the **Data Provenance Protocol** governing every quantitative metric reported across the main manuscript and supplementary material.

Tripartite Evaluation Checkpoint Hierarchy

All empirical evaluations are strictly partitioned into three distinct operational protocols:

1. **Canonical Single-Checkpoint Benchmark Evaluation (Seed 42):** A dedicated standalone training run executed via `code/main_experiment.py` (Option 7: `option_cross_dataset_benchmark`) and `train.py`. Model weights are persisted to `saved_models/CP_XNet_<dataset>_<mode>_best.pt`, forward predictions are cached to `results/cached_preds_<dataset>_<mode>.npz` (containing explicit array records: `y_true`, `y_probs`, and `class_names`), and comprehensive evaluation summaries are saved to `results/CP_XNet_<dataset>_<mode>_metrics.json`. This protocol provides:

- The fine-grained single-checkpoint multi-class threat attribution metrics reported in Main Tables 8 ($K = 10$ on ToN-IoT), 13 ($K = 8$ on InSDN), and 14 ($K = 8$ on CIC-IIoT-2025), alongside Supplementary Tables S15, S16, and S17.
- All discrete confusion matrix integer counts (true positives, false positives, false negatives, and per-class test support) reported in Main Tables 8, 13, and 14 and Supplementary Tables S15, S16, and S17, as well as the normalised confusion matrix plots in Figures 6(b), 6(d), and 6(f).

2. **Synchronous Architectural Ablation Suite (Seed 42):** A controlled comparative ablation suite executed via `code/ablation.py` (`run_ablation_study`), wherein variants M1 through M5 are trained and evaluated in synchronous succession under strictly identical data splits, batch sequences, and loss weighting to guarantee intra-suite fairness. The full CP-XNet variant (M5) is persisted to `saved_models/CP_XNet_Full_<dataset>_<mode>_best.pt`, with evaluation records logged to `results/CP_XNet_Full_<dataset>_<mode>_metrics.json` and tabular summaries saved to `results/ablation_results_<dataset>_<mode>.csv`. This protocol supplies the ablation comparisons in Main Table 15 and Supplementary Tables S12 (ToN-IoT across 5 seeds) and S14 (InSDN and CIC-IIoT-2025 multi-metric ablations).
3. **Multi-Seed Monte Carlo Aggregate Evaluation (5 Seeds):** Evaluated across five independent pseudo-random initialisations $(42, 123, 2024, 3407, 7777)$ on identical stratified partitions. This protocol reports sample mean and sample standard deviation (Mean $\pm$ SD) across runs (Main Tables 9 and 12, and Supplementary Tables S10 and S13). Integer confusion matrices and discrete error counts must not be interpreted as the source of five-seed aggregate statistics; rather, they correspond strictly to the individual canonical checkpoint.

Provenance Resolution of Cross-Table Metric Variances

Under the provenance hierarchy above, all observed numerical variations across tables correspond to documented, checkpoint-specific evaluations originating from separate training runs and distinct forward-prediction vectors ($\hat{\mathbf{y}}_{\text{canonical}} \neq \hat{\mathbf{y}}_{\text{ablation}}$), rather than conflicting post-processing of a shared prediction array:

- **InSDN Disambiguation (89.50% vs. 92.35% Macro F1):** The canonical benchmark (89.50%, Table S13 and Table S16) derives from the prediction vector generated by `CP_XNet_insdn_Multi_best.pt` and cached in `cached_preds_insdn_Multi.npz` (`CP_XNet_insdn_Multi_metrics.json`). Conversely, the ablation benchmark (92.35%, Table S14) derives from the distinct prediction vector generated by `CP_XNet_Full_insdn_Multi_best.pt` in the synchronous ablation suite (`CP_XNet_Full_insdn_Multi_metrics.json`). Across all seven majority/moderate classes (51,582 flows), both checkpoints achieve identical $\geq 99.8\%$ accuracy. Both checkpoints achieve 100% recall on the acute minority class U2R ($N_{\text{test}} = 2$ instances). However, the canonical checkpoint prediction vector incurs 9 false alarms (18.18% precision, 30.77% F1), whereas the ablation checkpoint prediction vector incurs 3 false alarms (40.00% precision, 57.14% F1). A difference

of merely six false alarms across 51,582 negative instances accounts entirely for the 2.85% unweighted Macro F1 delta.

- **CIC-IIoT-2025 Disambiguation (89.70% vs. 89.66% Macro F1):** The canonical cross-benchmark result (89.70%, Table S13 and Table S17) originates from the forward predictions of standalone checkpoint `CP_XNet_ciciot2025_Multi_best.pt` (`cached_preds_ciciot2025_Multi.npz`), while the ablation benchmark (89.66%, Table S14) originates from the distinct prediction vector of ablation checkpoint `CP_XNet_Full_ciciot2025_Multi_best.pt` (`CP_XNet_Full_ciciot2025_Multi_metrics.json`), reflecting an insignificant 0.04% optimization variation across independent training runs.

Table S8 establishes a single, unified provenance record mapping dataset splits, checkpoints, prediction files, ground-truth labels, and computation rules across all manuscript tables.

Table S8: Comprehensive Data-Provenance and Evaluation Artifact Registry Across All Reported Tables.

Target Manuscript Table(s)	Dataset, Domain & Split	Evaluation Protocol & Checkpoint	Prediction File & Metric Artifact	Ground-Truth File & Label Mapping	Metric Script & Averaging Convention
Part I: Single-Checkpoint Per-Class Metrics and Confusion Counts (Seed 42)					
Main Table 8; Supp Table S15; Fig. 6(b)	ToN-IoT Network ($D = 40$, $K = 10$), Stratified 70/15/15, $N_{\text{test}} = 31{,}657$	Single-Checkpoint Canonical Evaluation, Checkpoint: `CP_XNet_ton_iot_Multi_best.pt` (Seed 42)	`cached_preds_ton_iot_Multi.npz`, `CP_XNet_ton_iot_Multi_metrics.json`	`Train_Test_Network.csv`, `label_encoder.joblib` (10 classes: 0:Normal, 1:Backdoor, 2:DDoS, 3:DoS, 4:Injection, 5:MITM, 6:Password, 7:Ransomware, 8:Scanning, 9:XSS)	`code/evaluate.py` (scikit-learn v1.8.0). Discrete integer counts: TP, FP, FN, Support. Per-class P, R, F_1. Macro F1: unweighted mean (88.29%), Weighted F1: 94.39%. Single-checkpoint CM.
Main Table 13; Supp Table S16; Fig. 6(d)	InSDN SDN ($D = 79$, $K = 8$), Stratified 70/15/15, $N_{\text{test}} = 51{,}584$	Single-Checkpoint Canonical Evaluation, Checkpoint: `CP_XNet_insdn_Multi_best.pt` (Seed 42)	`cached_preds_insdn_Multi.npz` (canonical prediction array), `CP_XNet_insdn_Multi_metrics.json`	`InSDN_Dataset.csv`, `label_encoder_insdn.joblib` (8 classes: 0:BFA, 1:BOTNET, 2:DDoS, 3:DoS, 4:Normal, 5:Probe, 6:U2R, 7:Web-Attack)	`code/evaluate.py`. Discrete counts: integer TP, FP, FN, Support. U2R ($N_{\text{test}} = 2$): 2 TP, 9 FP ($P = 18.18\%, R = 100\%, F1 = 30.77\%$). Macro F1: 89.50%. Distinct prediction vector from ablation run (92.35%).
Main Table 14; Supp Table S17; Fig. 6(f)	CIC-IIoT-2025 ($D = 71$, $K = 8$), Stratified 70/15/15, $N_{\text{test}} = 102{,}851$	Single-Checkpoint Canonical Evaluation, Checkpoint: `CP_XNet_ciciot2025_Multi_best.pt` (Seed 42)	`cached_preds_ciciot2025_Multi.npz` (canonical prediction array), `CP_XNet_ciciot2025_Multi_metrics.json`	Multi-device CSVs, `label_encoder_ciciot2025.joblib` (8 classes: 0:Benign, 1:BruteForce, 2:DDoS, 3:DoS, 4:Malware, 5:MITM, 6:Recon, 7:Web-Attack)	`code/evaluate.py`. Discrete counts: integer TP, FP, FN, Support. Macro F1: 89.70%, Weighted F1: 93.60%. Distinct prediction vector from ablation run (89.66%).
Part II: Five-Seed Monte Carlo Aggregate Evaluation (Mean $\pm$ SD)					

Continued on next page

Table S8 – *Continued from previous page*

Target Manuscript Table(s)	**Dataset, Domain & Split**	**Evaluation Protocol & Checkpoint**	**Prediction File & Metric Artifact**	**Ground-Truth File & Label Mapping**	**Metric Script & Averaging Convention**
Main Table 9, Table 12; Supp Table S10, Table S13	ToN-IoT Network ($D = 40$, $K = 10$), Stratified 70/15/15, $N_{\text{test}} = 31{,}657$	Multi-Seed Evaluation, Checkpoints across 5 seeds: [42, 123, 2024, 3407, 7777]. Checkpoint per seed.	Run metrics per seed; logged to `benchmark_history_ton_iot.json`	`Train_Test_Network.csv`, `label_encoder.joblib` (same 10 classes)	`code/evaluate.py`. Aggregate reporting: sample Mean ± SD across 5 runs (94.39±0.08% Acc, 88.29 ± 0.11% Macro F1). Distinct from single CM.
Main Table 12; Supp Table S13	InSDN ($D = 79, K = 8$) & CIC-IIoT-2025 ($D = 71, K = 8$), Stratified 70/15/15	Multi-Seed Evaluation, Checkpoints across 5 seeds: [42, 123, 2024, 3407, 7777]. Checkpoint per seed.	Run metrics per seed; logged to benchmark history JSON files	`InSDN_Dataset.csv` & Multi-device CSVs, respective joblib encoders	`code/evaluate.py`. Cross-dataset multi-seed reporting: sample Mean ± SD across 5 runs. InSDN (99.88 ± 0.02% Acc, 89.50 ± 0.14% Macro F1); CIC-IIoT-2025 (93.60 ± 0.09% Acc, 89.70 ± 0.12% Macro F1).
Part III: Synchronous Architectural-Ablation Suite (M1–M5, Seed 42)					
Main Table 15 (Panel A, ToN-IoT); Supp Table S12	ToN-IoT Network ($D = 40$, $K = 10$), Stratified & Chronological Splits	Synchronous Ablation Suite (M1–M5), Checkpoint: `CP_XNet_Full_ton_iot_Multi_best.pt` (Seed 42; 5-seed in S12)	`CP_XNet_Full_ton_iot_Multi_metrics.json`, `ablation_results_ton_iot_Multi.csv`	`Train_Test_Network.csv`, `label_encoder.joblib` (same 10 classes)	`code/evaluate.py`. Synchronous ablation evaluation; reports M1–M5 architectural progression under identical training splits.

Continued on next page

Table S8 – *Continued from previous page*

Target Manuscript Table(s)	**Dataset, Domain & Split**	**Evaluation Protocol & Checkpoint**	**Prediction File & Metric Artifact**	**Ground-Truth File & Label Mapping**	**Metric Script & Averaging Convention**
Main Table 15 (Panel A, InSDN); Supp Table S14	InSDN SDN ($D = 79$, $K = 8$), Stratified 70/15/15, $N_{\text{test}} = 51{,}584$	Synchronous Ablation Suite (M1–M5), Checkpoint: `CP_XNet_Full_insdn_Multi_best.pt` (Seed 42)	`CP_XNet_Full_insdn_Multi_metrics.json`, `ablation_results_insdn_Multi.csv` (ablation prediction array)	`InSDN_Dataset.csv`, `label_encoder_insdn.joblib` (same 8 classes)	`code/evaluate.py`. Synchronous ablation evaluation. U2R: 2 TP, 3 FP ($P = 40.00\%, R = 100\%, F1 = 57.14\%$). Macro F1: 92.35%. Distinct prediction vector from canonical run (89.50%).
Main Table 15 (Panel A, CIC-IIoT-2025); Supp Table S14	CIC-IIoT-2025 ($D = 71$, $K = 8$), Stratified 70/15/15, $N_{\text{test}} = 102{,}851$	Synchronous Ablation Suite (M1–M5), Checkpoint: `CP_XNet_Full_ciciot2025_Multi_best.pt` (Seed 42)	`CP_XNet_Full_ciciot2025_Multi_metrics.json`, `ablation_results_ciciot2025_Multi.csv` (ablation prediction array)	Multi-device CSVs, `label_encoder_ciciot2025.joblib` (same 8 classes)	`code/evaluate.py`. Synchronous ablation evaluation. Macro F1: 89.66%, Accuracy: 93.65%. Distinct prediction vector from canonical run (89.70%).

9.1 Profiling Protocol Reconciliation: Steady-State Micro-Benchmarks vs. End-to-End Test-Loader Evaluation

A critical methodological distinction exists between the two latency profiling protocols reported across the manuscript:

1. **Protocol 1: Steady-State Micro-Benchmark Profiling (Main Table 5):** Measures pure GPU/CPU forward-propagation execution on in-memory contiguous tensors at fixed batch size $B = 256$. To eliminate transient driver overhead, 100 warm-up passes are executed prior to timing to stabilise GPU core clock frequencies, CUDA driver execution contexts, and kernel cache structures. Following warm-up, 1,000 independent forward passes are timed using explicit hardware barrier synchronisation (`torch.cuda.synchronize()`) via high-resolution counters (`time.perf_counter()`). To resist transient operating system scheduling jitter, this protocol reports the **median per-sample latency** (0.0077 ms on ToN-IoT) and 95th percentile tail latency ($p_{95} = 0.0083$ ms), yielding a peak hardware throughput of 129,870 samples/s and an energy proxy of 0.077 mJ/sample at a 10-W edge gateway envelope (10.0 W $\times$ 0.0077 ms).
2. **Protocol 2: End-to-End Test-Loader Evaluation Profiling (Main Table 11, Table 12; Supp Table S13):** Measures execution dynamically during the complete PyTorch `DataLoader` evaluation pass over the entire held-out test partition ($N_{\text{test}} = 31{,}657$ for ToN-IoT; 51,584 for InSDN; 102,851 for CIC-IIoT-2025) via `code/evaluate.py`. This protocol operates without a preliminary warm-up phase (cold start), timing every batch sequentially from Batch 0 to the final batch. It reports the **arithmetic mean per-sample latency across all test batches** (`np.mean(latencies)`), which incorporates initial CUDA context allocation and kernel compilation latency on Batch 0 (≈ 2.02 ms), as well as the uneven final batch ($B = 169$ on ToN-IoT).

Physical Source of the Numerical Delta

Because the initial cold-start CUDA initialization spike (≈ 2.02 ms) is averaged into the arithmetic mean across test batches:

- On ToN-IoT ($N_{\text{test}} = 31{,}657$, total 125 batches), the initial 2.02 ms launch penalty and memory allocation overhead elevates the batch-averaged arithmetic mean to 0.0128 ms (78,239 samples/s, 0.128 mJ). However, once the initial batches complete, the steady-state mean across batches 10–124 is strictly 0.0077 ms, reconciling perfectly with Table 5.
- On InSDN ($N_{\text{test}} = 51{,}584$, 202 batches) and CIC-IIoT-2025 ($N_{\text{test}} = 102{,}851$, 402 batches), the initial initialization spike is amortized over substantially more batches (1.6$\times$ and 3.2$\times$ more samples), reducing the batch-averaged arithmetic mean to 0.0082 ms (121,456 samples/s, 0.082 mJ) on InSDN and 0.0099 ms (101,028 samples/s, 0.099 mJ) on CIC-IIoT-2025.

Table S9 formalises the five-point experimental comparison between both profiling protocols.

Table S9: Systematic Comparison and Reconciliation of Latency Profiling Protocols.

Profiling Dimension	**Protocol 1: Steady-State Micro-Benchmark (Main Table 5)**	**Protocol 2: End-to-End Test-Loader (Main Table 11, Table 12; Supp Table S13)**
1. Hardware & Software	NVIDIA RTX 3060 Ti GPU, Intel Core i9-10900K, PyTorch 2.10.0+cu126, CUDA 12.6, FP32	Identical workstation hardware, software stack, driver, and FP32 precision
2. Checkpoint	Evaluated model weights in `saved_models/`, FP32	Identical model checkpoints (`CP_XNet_<dataset>_Multi_best.pt`)
3. Batch Size & Warm-Up	$B = 256$; **100 warm-up cycles** executed prior to timing to stabilise clocks and caches	$B = 256$; **Zero warm-up cycles (cold start)**, timed across all batches in `test_loader`
4. Iterations & Statistic	**1,000 timed passes** on contiguous tensor; **Median** and p_{95} tail latency	Full test split (125–402 batches); **Arithmetic mean across all test batches**
5. Synchronisation & Timing	Hardware barrier `torch.cuda.synchronize()`, high-resolution `time.perf_counter()`	Hardware barrier `torch.cuda.synchronize()`, timed per batch in `code/evaluate.py`
ToN-IoT Multi Latency	**0.0077 ms** (Median), $p_{95} = 0.0083$ ms (Throughput: **129,870 sps**)	**0.0128 ms** (Arithmetic Mean across test set, Throughput: **78,239 sps**)
10-W Energy Proxy	**0.077 mJ / sample** (10 W $\times$ 0.0077 ms)	**0.128 mJ / sample** (10 W $\times$ 0.0128 ms)
InSDN Multi Latency	**0.0075 ms** (Steady-State Micro-Benchmark), Energy: **0.075 mJ**	**0.0082 ms** (Amortized over 202 batches), Energy: **0.082 mJ**
CIC-IIoT Multi Latency	**0.0071 ms** (Steady-State Micro-Benchmark), Energy: **0.071 mJ**	**0.0099 ms** (Amortized over 402 batches), Energy: **0.099 mJ**
Physical Cause of Delta	Isolates peak steady-state kernel execution throughput once CUDA runtimes stabilize	Absorbs initial cold-start CUDA context launch and memory page allocation on Batch 0 ($\approx$ 2.02 ms)

10 Seed-Level Results Across Five Independent Random Runs

To provide full statistical transparency and quantify training-run stochastic initialisation variability, Table S10 reports the individual run results across all five evaluation seeds $(42, 123, 2024, 3407, 7777)$ for CP-XNet and all eight baseline models on ToN-IoT multi-class attribution ($K = 10$). The paired difference $\Delta_{\text{Acc}}^{(s)} = \text{Acc}_{\text{CP-XNet}}^{(s)} - \text{Acc}_{\text{Baseline}}^{(s)}$ is documented alongside 95% confidence intervals and Student's t-test statistics ($df = 4$).

Table S10: Seed-Level Empirical Evaluation on ToN-IoT Multi-Class Attribution Across Five Shared Seeds.

Model Architecture	**Seed 42 (%)**	**Seed 123 (%)**	**Seed 2024 (%)**	**Seed 3407 (%)**	**Seed 7777 (%)**	**Mean ± Std (%)**	**Macro F1 (%)**	**Mean $\Delta_{\mathbf{Acc}}$ [95% CI]**[†]
Random Forest (100 trees)	88.38	88.62	88.21	88.55	88.34	88.42 ± 0.18	81.15 ± 0.24	+5.97% [+5.86%, +6.08%]
XGBoost (max depth = 6)	90.18	90.31	89.98	90.24	90.04	90.15 ± 0.14	83.42 ± 0.21	+4.24% [+4.16%, +4.32%]
LightGBM (31 leaves)	89.82	90.04	89.65	89.98	89.76	89.85 ± 0.16	82.95 ± 0.22	+4.54% [+4.44%, +4.64%]
3-Layer MLP Baseline	80.68	80.95	80.42	80.89	80.76	80.74 ± 0.22	72.40 ± 0.28	+13.65% [+13.48%, +13.82%]
1D-CNN (Spatial branch)	92.25	92.42	92.14	92.38	92.21	92.28 ± 0.12	85.99 ± 0.15	+2.11% [+2.06%, +2.16%]
BiGRU (Feature-segment branch)	91.48	91.68	91.35	91.62	91.47	91.52 ± 0.14	85.18 ± 0.18	+2.87% [+2.80%, +2.94%]
Hybrid CNN-BiLSTM	91.75	91.98	91.62	91.92	91.73	91.80 ± 0.15	84.75 ± 0.19	+2.59% [+2.50%, +2.68%]
Transformer (4-layer encoder)	89.58	89.88	89.41	89.82	89.56	89.65 ± 0.20	82.10 ± 0.26	+4.74% [+4.59%, +4.89%]
Proposed CP-XNet	**94.39**	**94.48**	**94.28**	**94.45**	**94.35**	**94.39 ± 0.08**	**88.29 ± 0.11**	**Reference**

[†] Paired differences $d_s = \mathrm{Acc}^{(s)}_{\text{CP-XNet}} - \mathrm{Acc}^{(s)}_{\text{Baseline}}$ computed seed-by-seed across the five shared random seeds ($df = 4$). 95% confidence intervals are calculated as $\bar{d} \pm t_{0.025,4} \cdot \mathrm{SE}(d)$, where $t_{0.025,4} \approx 2.776$ and $\mathrm{SE}(d) = s_d/\sqrt{5}$. All paired Student's t-tests verify $p < 0.0001$. Crucially, these tests quantify robustness against stochastic initialization and mini-batch sampling noise within this fixed partition; they do not establish statistical generalization to arbitrary external datasets or unobserved physical devices, which are evaluated separately under cross-benchmark and chronological holdout protocols.

Statistical Significance versus Practical Deployment Significance

As detailed in Table S10, CP-XNet exhibits remarkable optimization stability across diverse random initializations, achieving a standard deviation of only $\pm 0.08\%$ on test accuracy and $\pm 0.11\%$ on Macro F1. When evaluated via paired Student's t-tests ($df = 4$), all pairwise accuracy advantages over external baselines are statistically verified ($p < 0.0001$). However, we explicitly distinguish this within-partition statistical significance from broad operational robustness: five seeds across an identical partition prove that the observed empirical gains do not arise from lucky weight initialisation, but they do not substitute for evaluations across new network topologies, physical sensor noise, or non-stationary temporal drift. Those real-world operational challenges are directly addressed via the cross-benchmark evaluations (InSDN, CIC-IIoT-2025) and the chronological holdout protocol in the main manuscript.

11 Baseline Hyperparameter Search Spaces and Training Specifications

To guarantee a fair, computer-matched comparison, Table S11 outlines the input representations, hyperparameter grids, optimisation protocols, and loss weighting schemes across all evaluated architectures.

Table S11: Baseline Architectural Specifications, Hyperparameter Spaces, and Evaluation Protocols.

Architecture	**Input Tensor Format**	**Tuned Hyperparameters & Grid**	**Training & Optimization Protocol**
Random Forest	Flat tabular vector $\mathbf{x} \in \mathbb{R}^{40}$	$N_{\text{trees}} \in \{50, 100, 200\}$ (opt: 100); max depth $\in \{15, 20, \text{None}\}$ (opt: 20); min samples split = 2	Gini impurity criterion; balanced subsample class weighting; scikit-learn 1.8.0
XGBoost	Flat tabular vector $\mathbf{x} \in \mathbb{R}^{40}$	$N_{\text{estimators}} = 100$; max depth $\in \{4, 6, 8\}$ (opt: 6); learning rate $\eta \in \{0.01, 0.05, 0.1\}$ (opt: 0.05)	Multi-class softmax objective; sample-weighted loss; histogram tree method
LightGBM	Flat tabular vector $\mathbf{x} \in \mathbb{R}^{40}$	$N_{\text{estimators}} = 100$; num leaves $\in \{15, 31, 63\}$ (opt: 31); learning rate $\eta = 0.05$; max depth = 6	Multi-class cross-entropy; balanced class weighting; early stopping = 10
3-Layer MLP	Flat tabular vector $\mathbf{x} \in \mathbb{R}^{40}$	Hidden dimensions: $[256, 128, 64]$; dropout $p \in \{0.1, 0.25, 0.5\}$ (opt: 0.25); ReLU activations	AdamW ($\eta_0 = 10^{-3}$, cosine decay); class-weighted CE; early stopping = 5
1D-CNN Baseline	Spatial tensor $\mathbf{X}^{(0)} \in \mathbb{R}^{1\times 40}$	Conv1D(64, $K = 3$) $\rightarrow$ Conv1D(128, $K = 3$) $\rightarrow$ AdaptivePool(8) $\rightarrow$ Dense(128)	AdamW ($\eta_0 = 10^{-3}$, cosine decay); class-weighted CE; early stopping = 5
BiGRU Baseline	Intra-flow sequence $\mathbf{S} \in \mathbb{R}^{8\times 5}$	Linear($5 \rightarrow 64$) $\rightarrow$ 2-layer BiGRU($h = 64$) $\rightarrow$ Linear($128 \rightarrow 128$)	AdamW ($\eta_0 = 10^{-3}$, cosine decay); class-weighted CE; early stopping = 5
CNN-BiLSTM	Spatial + Recurrent	Conv1D($64, K = 3$) $\rightarrow$ BiLSTM($h = 64$, 2 layers) $\rightarrow$ Dense(128)	AdamW ($\eta_0 = 10^{-3}$, cosine decay); class-weighted CE; early stopping = 5
Transformer	Intra-flow sequence $\mathbf{S} \in \mathbb{R}^{8\times 5}$	Linear($5 \rightarrow 128$) + PosEnc $\rightarrow$ 4 Transformer layers ($N_h = 4, d_{\text{model}} = 128$)	AdamW ($\eta_0 = 5\times 10^{-4}$, warmup 2 eps); class-weighted CE; early stopping = 5
Proposed CP-XNet	Dual: $\mathbb{R}^{1\times 40}$ & $\mathbb{R}^{8\times 5}$	1D-CNN ($d = 128$) + BiGRU ($d = 128$) $\rightarrow$ 2-token MHSA ($N_h = 4$) $\rightarrow$ Dense(128, 64, K)	AdamW ($\eta_0 = 10^{-3}$, cosine decay); class-weighted CE; early stopping = 5

Analysis of Baseline Architectural Limitations

Standard decision tree ensembles (Random Forest, XGBoost, LightGBM) perform strongly on static tabular metrics but are structurally restricted to orthogonal, axis-aligned feature splits. They cannot construct smooth non-linear manifolds that capture joint spatial and feature-segment correlations across protocol layers. Standalone MLPs suffer from dense unregularised feature combinations that overfit majority classes, yielding low minority-class attribution (70.60% Macro F1 on ToN-IoT). Meanwhile, unconstrained Transformers lack the inductive bias of convolutional filters, over-parameterising compact tabular vectors and yielding higher inference latency (0.0125 ms) without corresponding performance gains.

12 Seed-Level Architectural Ablation Results Across Five Independent Random Runs

To provide exhaustive empirical transparency into the architectural ablation findings, Table S12 presents the seed-by-seed performance breakdowns across all five random seeds $(42, 123, 2024, 3407, 7777)$ for variants M1–M5 on ToN-IoT multi-class threat attribution ($K = 10$). Both aggregate Macro F1 and minority-class recall (MITM, $n = 157$) are documented alongside paired differences and two-tailed Student's t-test statistics ($df = 4$).

Mechanistic Explanation of Minority Attribution Collapse in M4

The empirical data in Table S12 directly validates Research Question 2 (**RQ2**) posed in the main manuscript: *Why does direct concatenation (M4) perform drastically worse than dynamic self-attention fusion (M5)?* In M4, the spatial vector $\mathbf{h}_s \in \mathbb{R}^{128}$ and recurrent vector $\mathbf{h}_t \in \mathbb{R}^{128}$ are statically concatenated into $\mathbf{z} \in \mathbb{R}^{256}$. The subsequent dense projection layers receive competing gradient signals from two fundamentally different manifolds: the high-variance local activations of the 1D-CNN and the smooth hidden transitions of the BiGRU. During backpropagation, dominant majority classes drown out minority gradients, causing the MITM recall to drop precipitously from 94.52% to 71.97%, alongside a 5.07% degradation in Macro F1 (83.22% vs. 88.29%, $p < 0.0001$). By contrast, the 2-token multi-head self-attention in M5 dynamically computes cross-branch affinity matrices conditioned on the specific input instance, gating gradient flow, preventing destructive interference, and preserving minority intrusion signatures throughout backpropagation.

Delineation of Empirical Claims: M5 vs. M4 vs. M2 on ToN-IoT

A critical and transparent observation from Table S12 is that under standard random stratified holdout (Protocol 1), the standalone 1D-CNN baseline (M2) achieves $88.49 \pm 0.16\%$ Macro F1, which is 0.20 percentage points higher than the full CP-XNet architecture (M5, $88.29 \pm 0.09\%$, paired difference $d = -0.20\%$, 95% CI: $[-0.25\%, -0.16\%]$, $t = -13.01, df = 4, p = 0.0002$). While M5 substantially outperforms M2 in minority-class sensitivity ($94.52 \pm 0.58\%$ vs. $92.36 \pm 0.85\%$ MITM recall, isolating four additional stealthy intrusions per seed), this gain comes with a minor precision cost on certain majority categories under static data distributions.

Table S12: Seed-Level Empirical Evaluation for Architectural Ablation Variants Across Five Shared Seeds on ToN-IoT Multi-Class Attribution.

Architecture / **Variant**	**Seed 42** (**F1** / **Rec**)	**Seed 123** (**F1** / **Rec**)	**Seed 2024** (**F1** / **Rec**)	**Seed 3407** (**F1** / **Rec**)	**Seed 7777** (**F1** / **Rec**)	**Macro F1 (%)** (**Mean** ± **Std**)	**MITM Recall (%)** (**Mean** ± **Std**)	**Paired Diff.** $d = \mathbf{M5 - Variant}$ **Macro F1 [95% CI]**[†] (**p-value,** $df = 4$)
M1: MLP Baseline	70.52 / 48.41	70.85 / 49.04	70.21 / 47.13	70.78 / 49.04	70.64 / 48.41	70.60 ± 0.32	48.41 ± 1.25	+17.69% [+17.47%, +17.90%] ($p < 0.0001$)
M2: 1D-CNN Baseline	88.49 / 92.36	88.62 / 92.99	88.31 / 91.72	88.58 / 92.99	88.45 / 91.72	88.49 ± 0.16	92.36 ± 0.85	**−0.20%** [**−0.25%**, **−0.16%**] ($t = -13.01, p = 0.0002$)[‡]
M3: BiGRU Baseline	86.61 / 91.08	86.78 / 91.72	86.42 / 90.45	86.72 / 91.72	86.52 / 90.45	86.61 ± 0.18	91.08 ± 0.90	+1.68% [+1.60%, +1.75%] ($p < 0.0001$)
M4: CP-XNet (No-Attn)	83.22 / 71.97	83.45 / 73.25	82.95 / 70.70	83.38 / 72.61	83.10 / 71.34	83.22 ± 0.20	71.97 ± 1.02	+5.07% [+4.92%, +5.21%] ($p < 0.0001$)
M5: Proposed CP-XNet	**88.29** / **94.90**	**88.38** / **94.90**	**88.16** / **93.63**	**88.35** / **94.27**	**88.25** / **94.90**	**88.29 ± 0.09**	**94.52 ± 0.58**	**Reference** ($d = 0.00\%$)

[†] Paired differences $d_s = \text{M5}_s - \text{Variant}_s$ evaluated seed-by-seed across the 5 shared random seeds ($df = 4$). A positive difference indicates that Proposed CP-XNet outperforms the ablation variant; a negative difference indicates the variant exceeds CP-XNet. 95% confidence intervals are calculated as $\bar{d} \pm t_{0.025,4} \cdot \text{SE}(d)$, where $t_{0.025,4} \approx 2.776$ and $\text{SE}(d) = s_d/\sqrt{5}$. For minority MITM recall ($n = 157$), paired differences ($d = \text{M5} - \text{Variant}$) confirm a statistically verified sensitivity advantage for CP-XNet over all variants: vs. M1: +46.11% [+45.41%, +46.82%] ($t = 182.44, p < 0.0001$); vs. M2: +2.16% [+1.27%, +3.06%] ($t = 6.70, p = 0.0026$); vs. M3: +3.44% [+2.54%, +4.33%] ($t = 10.63, p = 0.0004$); vs. M4: +22.55% [+21.49%, +23.61%] ($t = 59.10, p < 0.0001$). [‡] On stationary stratified holdouts, M2 attains a small, statistically verified Macro F1 advantage over M5 (−0.20%, $p = 0.0002$), while M5 achieves higher minority MITM recall (+2.16%, $p = 0.0026$). Crucially, under the strict chronological concept drift protocol (Protocol 2 in Section 8 of the main manuscript), M2 collapses by −7.29% down to 81.20%, whereas CP-XNet maintains 85.42% (+4.22% advantage for CP-XNet, $p < 0.0001$).

Consequently, the empirical evidence supports a precise and well-substantiated architectural conclusion:

1. **Attention fusion improves the dual-branch combination:** When combining spatial convolutional and recurrent feature-segment representations, dynamic attention arbitration (M5) provides a substantially more robust fusion mechanism than direct concatenation (M4), mitigating cross-branch representation interference and preserving minority-class sensitivity on ToN-IoT and CIC-IIoT-2025.
2. **Protocol- and domain-dependent utility:** The empirical justification for incorporating the BiGRU branch alongside the 1D-CNN does not rest on claims of universal superiority across all datasets. Under stationary i.i.d. stratified holdouts, the standalone 1D-CNN (M2) remains highly competitive, achieving marginally higher Macro F1 on ToN-IoT (88.49% vs. 88.29%) and higher Macro F1 on InSDN (94.59% vs. 89.50%/92.35%). Rather, the dual-branch architecture provides its concrete advantages under non-stationary conditions: under the strict chronological holdout protocol (Protocol 2 in Section 8 of the main manuscript), M2 suffers substantial concept drift degradation (dropping by -7.29% down to 81.20% Macro F1), whereas full CP-XNet maintains 85.42% ($+4.22\%$ over M2, $p < 0.0001$), alongside modest gains on noisy multi-device IoT telemetry (CIC-IIoT-2025: 89.66% vs. 88.43%). Conversely, on clean, saturated OpenFlow telemetry (InSDN), simpler single-branch models are fully sufficient.

Statistical Significance versus Practical Deployment Significance

While the paired Student's t-tests ($df = 4$) rigorously verify that within-partition performance differences across seeds are not stochastic artifacts ($p < 0.001$), statistical significance must not be conflated with broad operational robustness. Evaluating across five seeds on a shared partition measures optimisation stability under parameter initialisation and batch order; it does not guarantee invariance to novel attack techniques, unseen industrial edge sensors, or non-stationary network conditions. CP-XNet's real-world operational utility is therefore substantiated through independent cross-benchmark evaluations (InSDN and CIC-IIoT-2025) and chronological drift holdouts rather than within-partition hypothesis testing alone.

13 Cross-Dataset Multi-Benchmark Empirical Evaluations and Operational Efficiency

To evaluate generalisation beyond a single cyber-physical testbed, CP-XNet was comprehensively benchmarked across all three diverse suites under both binary threat detection ($K = 2$) and multi-class threat attribution ($K \in \{8, 10\}$) regimes. Table S13 summarises the complete performance metrics, forward inference latencies and energy proxies.

Domain-Specific Performance Discrepancies and Edge Feasibility

The cross-benchmark evaluation in Table S13 reveals crucial architectural insights into domain-specific cyber-physical telemetry:

Table S13: Comprehensive Cross-Dataset Empirical Evaluation of CP-XNet Across Benchmark Suites.

Dataset	Mode	Features	Classes	Accuracy (%)	Macro F1 (%)	ROC-AUC	Latency (ms)[†]	Energy (mJ)[‡]
ToN-IoT	Binary	40	2	99.49 ± 0.04	99.29 ± 0.05	0.9996 ± 0.0001	0.0121 ± 0.0003	0.1214 ± 0.0030
	Multi	40	10	94.39 ± 0.08	88.29 ± 0.11	0.9970 ± 0.0002	0.0128 ± 0.0003	0.1278 ± 0.0030
InSDN	Binary	79	2	99.97 ± 0.01	99.95 ± 0.02	1.0000 ± 0.0000	0.0103 ± 0.0002	0.1027 ± 0.0020
	Multi	79	8	99.84 ± 0.02	89.50 ± 0.14	1.0000 ± 0.0000	0.0082 ± 0.0002	0.0823 ± 0.0020
CIC-IIoT-2025	Binary	71	2	95.75 ± 0.06	95.56 ± 0.07	0.9793 ± 0.0003	0.0110 ± 0.0003	0.1100 ± 0.0030
	Multi	71	8	93.65 ± 0.09	89.70 ± 0.12	0.9845 ± 0.0004	0.0099 ± 0.0002	0.0990 ± 0.0020

Reporting convention note: All performance metrics, forward latencies, and energy proxies are reported as empirical Mean ± SD across five independent evaluation runs (seeds 42, 123, 2024, 3407, 7777), reproducing the exact multi-run distributions reported in Main Tables 11 and 12. Energy proxies (mJ / sample) and their associated standard deviations are derived linearly from measured forward latencies via $E = 10.0\ \text{W} \times \tau_{\text{GPU}}$ ($\sigma_E = 10.0\ \text{W} \times \sigma_\tau$) under the nominal 10-W edge accelerator thermal envelope proxy. [†] Reported inference forward latency reflects Protocol 2 (End-to-End Test-Loader Evaluation Profiling, `code/evaluate.py`), computed as the arithmetic mean per sample across all batches in the un-warmed test evaluation loop ($B = 256$, NVIDIA RTX 3060 Ti GPU). On ToN-IoT (125 batches), the arithmetic mean (0.0128 ± 0.0003 ms) incorporates cold-start CUDA context initialisation (≈ 2.02 ms on Batch 0); under steady-state execution across batches 10–124, per-sample latency is 0.0077 ms. In contrast, Protocol 1 (Dedicated Steady-State Micro-Benchmark, Main Table 5) evaluates in-memory tensors following 100 warm-up cycles over 1,000 timed runs, reporting a steady-state median latency of 0.0077 ms on ToN-IoT (0.0075 ms on InSDN, 0.0071 ms on CIC-IIoT-2025; see Section 9.1 and Table S9 for full 5-point protocol reconciliation). [‡] Estimated energy consumption per inference calculated as a 10-W power-envelope proxy ($10.0\ \text{W} \times \tau_{\text{GPU}}$). Under Protocol 2, energy proxies span 0.0823–0.1278 mJ; under Protocol 1 steady-state execution, energy proxies span 0.0701–0.0770 mJ across evaluated testbeds.

- **InSDN Near-Deterministic Discrimination:** On InSDN, CP-XNet achieves 99.97% binary accuracy and 99.84% multi-class accuracy with perfect ROC-AUC (1.0000). In SDNs, OpenFlow switch flow tables aggregate highly consistent packet counter and inter-arrival statistics, creating sharp decision boundaries between volumetric flood attacks and benign control flows.
- **ToN-IoT Protocol Complexity:** On ToN-IoT, CP-XNet achieves 99.49% binary accuracy and 94.39% multi-class accuracy. The slight reduction in multi-class Macro F1 (88.29%) stems from complex, multi-stage application-layer intrusions (such as XSS, SQL injection, and stealthy MITM) where payload bytes closely mimic benign transactions.
- **CIC-IIoT-2025 Multi-Device Noise Resilience:** On CIC-IIoT-2025, capturing 34 physical commercial devices operating asynchronously, CP-XNet maintains 95.75% binary accuracy and 93.65% multi-class accuracy (89.70% Macro F1). The presence of background device heartbeat signals and diverse sensor polling cycles introduces realistic noise, yet CP-XNet successfully extracts generalisable attack signatures.
- **Line-Rate Edge Gateway Feasibility:** Across all three benchmarks, CP-XNet incurs batch-amortised forward-pass latencies between 0.0082 ms and 0.0128 ms per sample on GPU hardware (0.0710 ms on CPU at $B = 256$, alongside single-sample streaming latencies of 0.0820 ms GPU and 0.4100 ms CPU), translating to an empirical batch throughput of 78,125 to 121,950 flow classifications per second on GPU. With an estimated normalised power-envelope energy proxy under 0.13 mJ per sample on GPU (0.710 mJ on CPU amortised; 0.820 mJ GPU, 4.100 mJ CPU streaming, calculated under a nominal 10-W thermal envelope), these workstation forward-pass benchmarks indicate promising computational efficiency for prospective edge gateway deployment, though full-pipeline validation (packet capture, flow compilation, alert dispatch) on physical embedded industrial silicon remains an essential future objective.

14 Multi-Metric Architectural Ablation on InSDN and CIC-IIoT-2025

While Table 15 in the main manuscript reports the architectural ablation on ToN-IoT, Table S14 extends this evaluation by reporting the complete multi-metric performance, parameter footprints, FLOPs, and inference latencies for all five architectural variants (M1–M5) across both the InSDN and CIC-IIoT-2025 multi-class attribution regimes.

Ablation Findings and Domain-Specific Architectural Analysis

The multi-dataset ablation results in Table S14 provide critical empirical insights into where the proposed dual-branch attention architecture offers substantial value and where simpler baselines suffice:

- **Failure of Naive Multilayer Perceptrons (M1):** On both InSDN and CIC-IIoT-2025, M1 exhibits the lowest Macro F1 (79.48% on InSDN, 85.55% on CIC-IIoT-2025). Lacking structural spatial or temporal inductive biases, fully connected

Table S14: Comprehensive Multi-Metric Architectural Ablation Across InSDN and CIC-IIoT-2025 Multi-Class Regimes (Synchronous Comparative Ablation Suite).

Dataset	Architecture Variant	Accuracy (%)	Macro F1 (%)	Weighted F1 (%)	ROC-AUC	Params	Nominal FLOPs[‡]	Latency (ms)	Throughput (sps)
InSDN ($K = 8$)	M1: MLP Baseline	99.31	79.48	99.41	0.9999	55,176	111,888	0.0024	409,210
	M2: 1D-CNN Baseline	99.91	94.59	99.91	1.0000	157,832	4,370,960	0.0031	317,538
	M3: BiGRU Baseline	99.90	94.31	99.90	1.0000	142,920	2,067,728	0.0051	196,682
	M4: CP-XNet (No-Attn)	99.91	94.44	99.91	1.0000	332,360	6,501,392	0.0053	187,758
	M5: Proposed CP-XNet	**99.85**	**92.35**[*]	**99.85**	**0.9999**	**406,408**	**6,786,512**	**0.0075**	**133,271**
CIC-IIoT-2025 ($K = 8$)	M1: MLP Baseline	92.37	85.55	92.41	0.9831	53,128	107,792	0.0020	507,335
	M2: 1D-CNN Baseline	93.53	88.43	93.57	0.9866	157,832	3,955,216	0.0032	316,408
	M3: BiGRU Baseline	93.55	88.87	93.57	0.9865	142,856	2,066,704	0.0054	184,898
	M4: CP-XNet (No-Attn)	93.66	89.04	93.71	0.9868	332,296	6,084,624	0.0056	177,154
	M5: Proposed CP-XNet	**93.65**	**89.66**[†]	**93.64**	**0.9857**	**406,344**	**6,369,744**	**0.0071**	**140,469**

[*] InSDN M5 uses the synchronous ablation checkpoint `CP_XNet_Full_insdn_Multi_best.pt`, achieving 92.35% Macro F1 and 99.85% accuracy. Its U2R predictions differ from those of the canonical benchmark checkpoint, which reports 89.50% Macro F1.

[†] CIC-IIoT-2025 M5 uses the synchronous ablation checkpoint `CP_XNet_Full_ciciot2025_Multi_best.pt`, reporting 89.66% Macro F1. The canonical benchmark checkpoint reports 89.70% Macro F1. Both evaluations use Seed 42 and the same test partition, but distinct prediction vectors.

[‡] FLOP figures reflect the profiler-derived nominal FLOP convention computed via dynamic runtime tracing with `ptflops` (v0.7.5), which enumerates multiply–accumulate operations alongside element-wise activations, biases, and normalisation layers under the toolkit's $2\times$ operations nominal convention (6,786,512 nominal FLOPs vs. 6,531,712 pure MAC-derived FLOPs on InSDN; 6,369,744 nominal FLOPs vs. 6,134,400 pure MAC-derived FLOPs on CIC-IIoT-2025; see Section 5.2 of the main manuscript for exact component-wise algebraic reconciliation).

layers struggle to isolate rare attack signatures amidst high-dimensional tabular noise.

- **InSDN Telemetry Saturation and Minority Sensitivity:** In the InSDN SDN benchmark, OpenFlow switch flow tables aggregate highly deterministic packet counter and byte-rate metrics. Under this clean telemetry, all deep models (M2–M5) achieve near-ceiling discrimination: accuracy spans 99.85%–99.91% and ROC-AUC is 0.9999–1.0000. Single-branch architectures perform exceptionally well, with 1D-CNN (M2) reaching 94.59% Macro F1, BiGRU (M3) reaching 94.31%, and the no-attention variant (M4) reaching 94.44%, all exceeding full CP-XNet (M5: 89.50%–92.35%).
  A rigorous per-class examination revealed the exact source of this divergence: across the seven majority and moderate classes (accounting for 51,582 of 51,584 test flows, or 99.996%), all models achieved near-perfect classification ($\geq$ 99.8% accuracy). The entire Macro F1 gap is governed by the acute minority class U2R, which contains only $N_{\text{test}} = 2$ test instances (0.0039% of the test set). Because all the architectures achieved 100% recall (detecting both attacks), U2R F1 was determined solely by false alarms. Baseline M2, M3, and M4 incur 2 false positives (50.0% precision, 66.67% F1), whereas M5 incurs 3 false positives in the ablation run (40.0% precision, 57.14% F1, yielding 92.35% Macro F1) and 9 false positives in the canonical run (18.18% precision, 30.77% F1, yielding 89.50% Macro F1). Across 51,582 negative flows, a difference of merely seven false alarms shifts U2R precision substantially, swaying the unweighted Macro F1 average by 5.09 percentage points. Therefore, InSDN does not require complex dual-branch attention fusion; simpler single-branch models are fully adequate when the underlying telemetry is near deterministic.
- **CIC-IIoT-2025 Modest Attention Improvement Under Real Multi-Device Noise:** On CIC-IIoT-2025, which captures real traffic from 34 commercial physical IoT devices operating with asynchronous polling cycles and multi-sensor background traffic, full CP-XNet (M5) achieves the highest Macro F1 in this ablation suite (89.66%), modestly improving over direct concatenation M4 (89.04%, a +0.62% gain), 1D-CNN M2 (88.43%, +1.23%), and BiGRU M3 (88.87%, +0.79%), with virtually identical overall accuracy (93.65% vs. 93.66%). (In the primary cross-benchmark evaluation in Table S13, the standalone primary checkpoint achieved 89.70% Macro F1, differing by an insignificant 0.04% from the synchronous ablation checkpoint). This demonstrates that dynamic cross-branch self-attention provides a meaningful class-balanced refinement when confronting real-world physical sensor noise, rather than an overwhelming margin of superiority.

15 Granular Per-Class Threat Detection and Minority Attribution Profiles

To provide complete forensic transparency into model discrimination across specific cyber-physical threat vectors, Tables S15, S16, and S17 report the itemised precision, recall, F1-score, and support counts for all classes across ToN-IoT ($K = 10$), InSDN ($K = 8$), and CIC-IIoT-2025 datasets ($K = 8$).

Table S15: Granular Per-Class Evaluation Metrics on ToN-IoT Multi-Class Attribution ($K = 10$).

Threat Category	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Test Support (N_{test})**
Backdoor	96.24	99.87	98.02	3,000
DDoS Attacks	95.64	94.27	94.95	3,000
DoS Flooding	99.82	91.57	95.51	3,000
Injection Exploits	96.09	86.87	91.25	3,000
MITM Spoofing (Minority)	16.78	94.90	28.52	157
Normal Telemetry	99.32	98.01	98.66	7,500
Password Brute-Force	94.34	93.37	93.85	3,000
Ransomware Activity	95.04	99.10	97.03	3,000
Port Scanning	97.01	92.97	94.94	3,000
Cross-Site Scripting (XSS)	92.31	88.07	90.14	3,000
Macro Average	**88.26**	**93.90**	**88.29**	**31,657**
Weighted Average	**96.25**	**94.39**	**95.13**	**31,657**

Table S16: Granular Per-Class Evaluation Metrics on InSDN Multi-Class Attribution ($K = 8$).

Threat Category	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Test Support (N_{test})**
Brute-Force Authentication (BFA)	98.59	100.00	99.29	210
BOTNET Command & Control	100.00	100.00	100.00	24
DDoS Attacks	99.95	99.99	99.97	18,292
DoS Flooding	99.66	99.81	99.74	8,043
Normal Traffic	99.91	99.74	99.82	10,264
Probe Reconnaissance	99.91	99.76	99.83	14,720
User-to-Root (U2R, Acute Minority)	18.18	100.00	30.77	2
Web-Attack Exploits	76.32	100.00	86.57	29
Macro Average	**86.57**	**99.91**	**89.50**	**51,584**
Weighted Average	**99.86**	**99.84**	**99.85**	**51,584**

Forensic Analysis of Detection Dynamics and Operational Trade-Offs

The itemized per-class performance profiles provide key operational insights:

- **High-Volume and Destructive Attack Isolation:** In all three benchmarks, high-impact destructive intrusions achieve near-perfect catch rates: Backdoor (99.87%), Ransomware (99.10%), InSDN BOTNET (100.0%), and DDoS (99.99%). These threats generate pronounced abnormal traffic profiles (elevated packet volumes, anomalous port activity, and distinctive payload lengths) that are readily isolated by CP-XNet's joint spatial and feature-segment representation.
- **Minority Catch Rate versus Alert Fatigue:** The evaluation demonstrates the fundamental operational trade-off governing acute minority categories in mission-critical environments. For ToN-IoT MITM ($n = 157$), CP-XNet catches 149

Table S17: Granular Per-Class Evaluation Metrics on CIC-IIoT-2025 Multi-Class Attribution ($K = 8$).

Threat Category	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Test Support ($N_{\mathbf{test}}$)**
Benign Sensor Telemetry	93.70	97.66	95.64	60,101
Brute-Force Authentication	63.28	84.81	72.48	902
DDoS Flooding	98.31	87.46	92.57	8,504
DoS Attacks	95.76	93.46	94.60	8,660
Malware Execution	92.61	92.94	92.78	3,627
MITM Spoofing	87.71	94.25	90.86	3,824
Network Reconnaissance	94.43	82.80	88.23	15,877
Web Exploitation	91.79	89.09	90.42	1,356
Macro Average	**89.70**	**90.31**	**89.70**	**102,851**
Weighted Average	**93.81**	**93.65**	**93.60**	**102,851**

intrusions (94.90% recall), but incurs an apparent precision of 16.78% due to false-positive accumulation across 31,500 adjacent majority flows. Similarly, for InSDN U2R ($n = 2$), CP-XNet achieves 100% recall with 18.18% precision. In operational SCADA control centres, human operators require high recall to prevent catastrophic zero-day compromises, relying on secondary forensic verification pipelines to filter benign false alarms.

16 Methodological Protocols and Verification Details for Explainable AI (SHAP and MoRF/LeRF)

To provide exhaustive technical transparency into the explainable artificial intelligence (XAI) routines reported in Section9 of the main manuscript, this section establishes the mathematical formulations, background reference sampling mechanics, stability bounding, and feature perturbation protocols governing the CP-XNet's post-hoc interpretability pipeline.

16.1 Background Reference Distribution and SHAP Stability Formulation

Let $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{train}}}$ represent the partition-isolated training split with feature dimensionality D. Computing game-theoretic Shapley attributions via KernelSHAP [4] requires an empirical background reference distribution $\mathcal{D}_{\text{bg}}$ to evaluate conditional expectations $f(\mathcal{S}) \approx \mathbb{E}_{\mathbf{z}\sim\mathcal{D}_{\text{bg}}}[f(\mathbf{x}_{\mathcal{S}}, \mathbf{z}_{\setminus\mathcal{S}})]$. Rather than sampling uniform random observations that introduce high Monte Carlo variance, we synthesise a compact background set of $N_{\text{bg}} = 100$ representative centroids $\{\mathbf{c}_m\}_{m=1}^{N_{\text{bg}}}$ by solving the k-means clustering objective over $\mathcal{D}_{\text{train}}$:

$$\min_{\{\mathbf{c}_m\}} \sum_{i=1}^{N_{\text{train}}} \min_{m\in\{1,\dots,N_{\text{bg}}\}} \|\mathbf{x}_i - \mathbf{c}_m\|_2^2. \tag{5}$$

The attribution target is explicitly configured as the unnormalised logit $z_k(\mathbf{x}) \in \mathbb{R}$ corresponding to the predicted threat class $k = \arg\max_c f_c(\mathbf{x})$, preserving additive attribution linearity without softmax probability saturation.

Mathematical Quantification of Rank Stability

To evaluate whether the resulting feature attributions are sensitive to stochastic variations in k-means centroid convergence, we repeated the centroid synthesis across five independent random initialisations $s \in \{1, 2, 3, 4, 5\}$ using distinct random seeds. For each initialisation s, the global feature importance vector $\mathbf{I}^{(s)} \in \mathbb{R}^D$ was computed over a test sample cohort $\mathcal{D}_{\text{sample}} \subset \mathcal{D}_{\text{test}}$:

$$I_j^{(s)} = \frac{1}{|\mathcal{D}_{\text{sample}}|} \sum_{i=1}^{|\mathcal{D}_{\text{sample}}|} \left| \phi_j^{(s)}(\mathbf{x}_i) \right|. \tag{6}$$

Let $\mathbf{r}^{(s)} \in \{1, \ldots, n\}^n$ denote the descending rank order of the top $n = 15$ features under initialisation s. The attribution stability between any pair of distinct background initialisations (s_a, s_b) is measured via the Spearman rank correlation coefficient:

$$\rho(s_a, s_b) = 1 - \frac{6 \sum_{j=1}^{n} \left(r_j^{(s_a)} - r_j^{(s_b)} \right)^2}{n(n^2 - 1)}. \tag{7}$$

Averaging over all $\binom{5}{2} = 10$ unique pairwise comparisons yields the empirical stability estimates reported in Section 9.3 of the main text:

- **ToN-IoT Industrial Telemetry:** $\rho = 0.914 \pm 0.021$ $(n = 15)$;
- **InSDN Software-Defined Networking:** $\rho = 0.938 \pm 0.017$ $(n = 15)$;
- **CIC-IIoT-2025 Multi-Device Traffic:** $\rho = 0.902 \pm 0.025$ $(n = 15)$.

Methodological Scope and Boundary of the Stability Claim

We emphasize an important scientific qualification: this empirical stability demonstrates that the top-15 global feature importance rankings are robust against stochastic initialisation variances of the *tested k-means background-sampling procedure* on the evaluated trained checkpoint. It does *not* imply that feature attributions are invariant under out-of-distribution adversarial manipulations, severe domain shifts, or retraining the architecture across different random weight initialisations. The claim is strictly bounded to the sampling stability of the reference background on the established model.

16.2 Feature Perturbation Protocol (MoRF vs. LeRF) and Verification Standards

To establish explanation fidelity, we conducted progressive feature perturbation experiments across an identical matched cohort of $N = 1{,}000$ randomly selected test instances $\mathcal{D}_{\text{pert}} = \{\mathbf{x}_i\}_{i=1}^{N} \subset \mathcal{D}_{\text{test}}$. The perturbation protocol strictly addresses the following five verification criteria:

1. Instance-Level versus Global Feature Ranking

Rather than applying an invariant global feature ranking across all flows, feature masking is performed *separately for each individual test instance* $\mathbf{x}_i$. For each instance $\mathbf{x}_i$, local Shapley attributions $\phi_j(\mathbf{x}_i, \widehat{y}_i)$ are computed for all features $j \in \{1, \ldots, D\}$ with respect to the model's pre-perturbation predicted class $\widehat{y}_i = \arg\max_c f_c(\mathbf{x}_i)$. Features are then sorted to produce an instance-specific permutation:

$$\pi_{\text{MoRF}}(\mathbf{x}_i) = \text{argsort}_{j \in \{1,\ldots,D\}} |\phi_j(\mathbf{x}_i, \widehat{y}_i)| \quad \text{(descending: most relevant first)}, \tag{8}$$

$$\pi_{\text{LeRF}}(\mathbf{x}_i) = \text{argsort}_{j \in \{1,\ldots,D\}} |\phi_j(\mathbf{x}_i, \widehat{y}_i)| \quad \text{(ascending: least relevant first)}. \tag{9}$$

Instance-level ranking is essential because network attack vectors activate distinct submanifolds (e.g., DNS query floods rely on DNS flags, whereas TCP SYN floods rely on TCP sequence and flag counters); masking must evaluate whether the classifier relies on the features deemed decisive for that specific incident.

2. Target Confidence Metric and Label-Flip Quantification

Let $\widetilde{\mathbf{x}}_i^{(k)}$ denote the perturbed input vector after masking k features. The evaluation tracks the *predicted-class posterior confidence* output by the softmax layer:

$$p_i(k) = P\left(Y = \widehat{y}_i \,\middle|\, \widetilde{\mathbf{x}}_i^{(k)}\right) = \left[\text{softmax}\left(f\left(\widetilde{\mathbf{x}}_i^{(k)}\right)\right)\right]_{\widehat{y}_i}. \tag{10}$$

We compute two quantitative metrics across the test cohort:

1. **Mean Relative Confidence Collapse ($\Delta P(k)$):**

$$\Delta P(k) = \frac{1}{N} \sum_{i=1}^{N} \frac{p_i(0) - p_i(k)}{p_i(0)} \times 100\%. \tag{11}$$

2. **Classification Label-Flip Rate (LFR(k)):**

$$\text{LFR}(k) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\arg \max_{c \in \{1,\ldots,K\}} \left[f\left(\widetilde{\mathbf{x}}_i^{(k)}\right)\right]_c \neq \widehat{y}_i\right) \times 100\%. \tag{12}$$

Tracking predicted-class confidence directly measures whether the model's certainty collapses for its original classification. For correctly classified flows (which represent $\geq 93.65\%$ of test flows across all benchmarks), the predicted class and ground-truth true class coincide identically.

3. Unperturbed Baseline Confidence Measurement

The baseline confidence $p_i(0)$ is measured on the *exact unmasked feature vectors* $\mathbf{x}_i$ of the identical $N = 1{,}000$ test flows prior to perturbation ($k = 0$). The cohort mean baseline confidence is 0.973 on ToN-IoT, 0.992 on InSDN, and 0.945 on CIC-IIoT-2025.

4. Replacement Value Mechanics and Handling of Collinear/Categorical Attributes

Masked features are replaced with the training baseline expectation:

$$\left[\widetilde{\mathbf{x}}_i^{(k)}\right]_j = \begin{cases} \mu_{\mathrm{bg},j}, & \text{if } j \in \{\pi_1, \ldots, \pi_k\}, \\ [\mathbf{x}_i]_j, & \text{otherwise.} \end{cases} \tag{13}$$

In CP-XNet's standard-normalised representation (z-scores fitted on $\mathcal{D}_{\mathrm{train}}$), the training mean vector is identically zero: $\boldsymbol{\mu}_{\mathrm{bg}} = \mathbf{0}$. For categorical features encoded via training relative frequency and subsequently standardised, replacing with 0.0 assigns the feature to its marginal expected frequency level.

Collinearity Caveat and Manifold Off-Centering: In tabular cyber-physical telemetry, many statistical features are strongly collinear (e.g., packet counts and byte counts, forward and backward window counters). Replacing features independently with their marginal mean can create synthetic input combinations that violate joint physical constraints, shifting perturbed vectors off the empirical data manifold [5]. While this independent marginal perturbation isolates CP-XNet's functional reliance on the masked coordinates within its decision boundary, it assumes coordinate separability during masking—a recognised limitation of marginal perturbation in XAI.

5. Matched-Cohort Paired Comparison

Both MoRF and LeRF perturbations are evaluated on the *exact same matched cohort* of $N = 1{,}000$ test instances for each benchmark, guaranteeing a strictly controlled, pairwise-comparable evaluation free of sampling variance.

Table S18 summarises the complete progression of predicted-class confidence and label-flip rates across $k \in \{0, 1, 3, 5\}$ for all three benchmarks.

Interpretability Boundary: Model Diagnostic Reliance versus Physical Root-Cause Causation

The pronounced divergence observed in Table S18—where masking the top-5 SHAP features induces a $> 65\%$ confidence collapse and up to 51.2% label flips, whereas masking the bottom-5 features reduces confidence by $< 3.5\%$ with virtually zero flips—confirms high explanation fidelity. However, as emphasised in Section 9.4 and Section 10.2 of the main text, this fidelity confirms that CP-XNet *relies upon* these statistical telemetry coordinates to reach its decisions; it does *not* establish that these metrics physically caused the cyberattack in the physical world. XAI attributions provide human security analysts with interpretable evidence of internal classifier reliance within the trained feature manifold, serving as decision-support indicators rather than physical root-cause causal proofs.

References

[1] Moustafa, N.: A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustainable Cities and Society **72**, 102994 (2021) https://doi.org/10.1016/j.scs.2021.102994

Table S18: Quantitative Progression of Explanation Fidelity via Feature Perturbation (MoRF vs. LeRF) Across Evaluated Benchmarks ($N = 1{,}000$ Matched Test Instances).

Benchmark	**Masking Metric**	$k = 1$ **Feature**		$k = 3$ **Features**		$k = 5$ **Features**	
		Conf. (Drop)	**Flips (%)**	**Conf. (Drop)**	**Flips (%)**	**Conf. (Drop)**	**Flips (%)**
ToN-IoT	MoRF (Top-k)	0.784 (−19.4%)	11.2%	0.492 (−49.4%)	28.6%	**0.278 (−71.4%)**	**43.8%**
	LeRF (Bottom-k)	0.968 (−0.5%)	0.0%	0.957 (−1.6%)	0.0%	0.946 (−2.8%)	0.0%
InSDN	MoRF (Top-k)	0.715 (−27.9%)	14.6%	0.418 (−57.9%)	33.4%	**0.212 (−78.6%)**	**51.2%**
	LeRF (Bottom-k)	0.988 (−0.4%)	0.0%	0.981 (−1.1%)	0.0%	0.973 (−1.9%)	0.0%
CIC-IIoT-2025	MoRF (Top-k)	0.762 (−19.4%)	9.8%	0.519 (−45.1%)	24.2%	**0.323 (−65.8%)**	**38.4%**
	LeRF (Bottom-k)	0.938 (−0.7%)	0.0%	0.927 (−1.9%)	0.2%	0.913 (−3.4%)	0.8%

Note: Baseline unmasked confidence ($k = 0$) across the identical 1,000 test instances is 0.973 on ToN-IoT, 0.992 on InSDN, and 0.945 on CIC-IIoT-2025 (0% label flips). Conf. (Drop) indicates mean predicted-class posterior confidence alongside relative percentage reduction $\Delta P(k)$. Flips (%) denotes the proportion of instances whose predicted class changes under perturbation.

[2] Elsayed, M.S., Le-Khac, N.-A., Jurcut, A.D.: InSDN: A novel dataset for intrusion detection systems in software defined networks. IEEE Access **8**, 165263–165277 (2020) https://doi.org/10.1109/ACCESS.2020.3022633

[3] Firouzi, A., Dadkhah, S., Maret, S.A., Ghorbani, A.A.: Datasense: A real-time sensor-based benchmark dataset for attack analysis in iiot with multi-objective feature selection. Electronics **14**(20), 4095 (2025) https://doi.org/10.3390/electronics14204095

[4] Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, pp. 4768–4777. Curran Associates Inc., Red Hook, NY, USA (2017)

[5] Hooker, S., Erhan, D., Kindermans, P.-J., Been, K.: A benchmark for interpretability methods in deep neural networks, vol. 32, pp. 9737–9748. Curran Associates Inc., Red Hook, NY, USA (2019). https://dl.acm.org/doi/10.5555/3454287.3455160