PANDORA: Lightweight Adversarial Defense For Edge IoT Using Uncertainty-Aware Metric Learning

Avinash Awasthi*, Pritam Vediya*, Hemant Miranka^, **Ramesh Babu Battula***, Manoj Singh Gaur

*Department of Computer Science and Engineering, Malaviya National Institute of Technology, Jaipur, India
^The LNM Institute of Information Technology, Jaipur, India
Indian Institute of Technology Jammu, Jammu and Kashmir, India

Funded by: Telecom Technology Development Fund Scheme
Department of Telecommunications (DoT), Government of India
Grant ID: TTDF/6G/517

Background & Motivation

The Evolving IoT Threat Landscape:

- Hyper-connectivity in Smart Cities, Industry 4.0/5.0, and Healthcare is driven by billions of resource-constrained devices.

- This expansion has dissolved traditional network perimeters, creating a massive, heterogeneous attack surface.

What helps to detect and alert the systems to keep secure from this evolving threats?

"Intrusion Detection Systems (IDS) *continuously monitor network and system activities, analyze suspicious behavior, and generate real-time alerts, enabling organizations to quickly identify, respond to, and mitigate potential security breaches before significant damage occurs."*

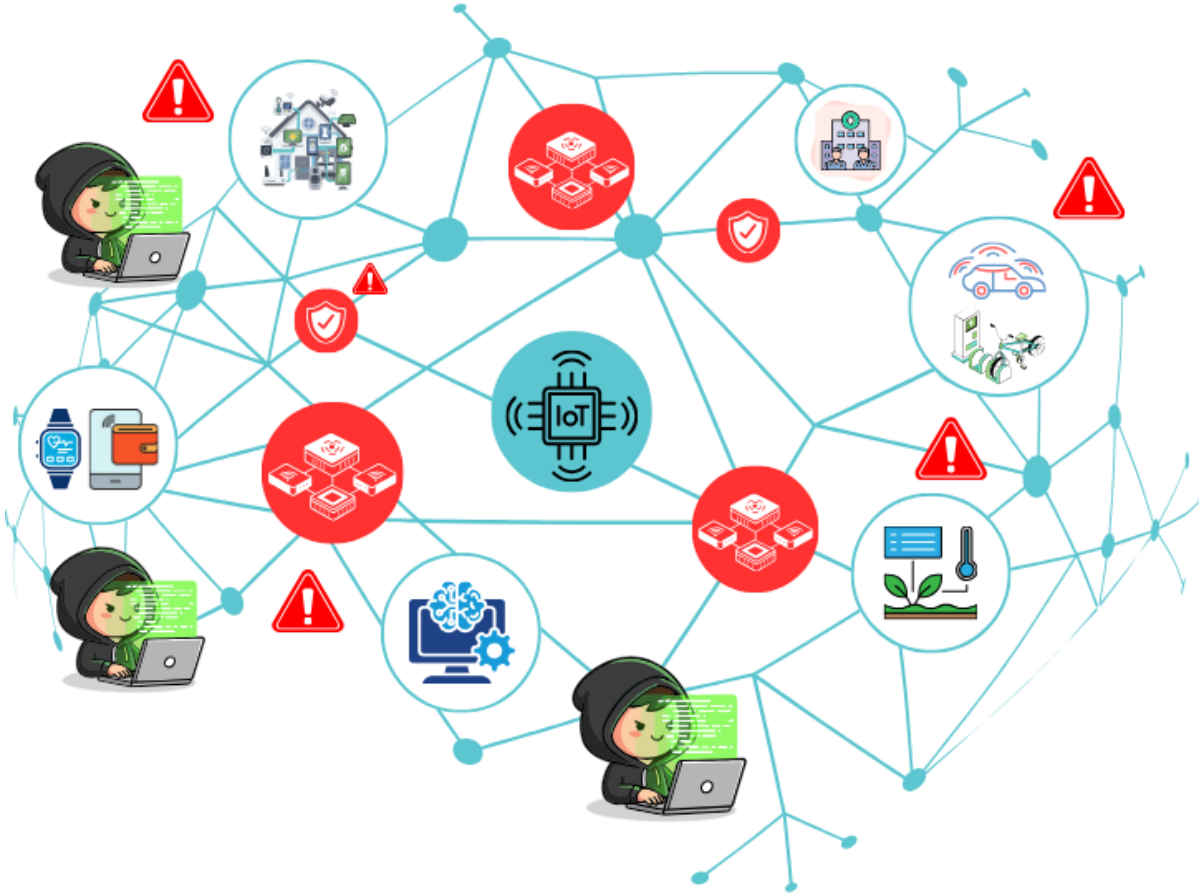
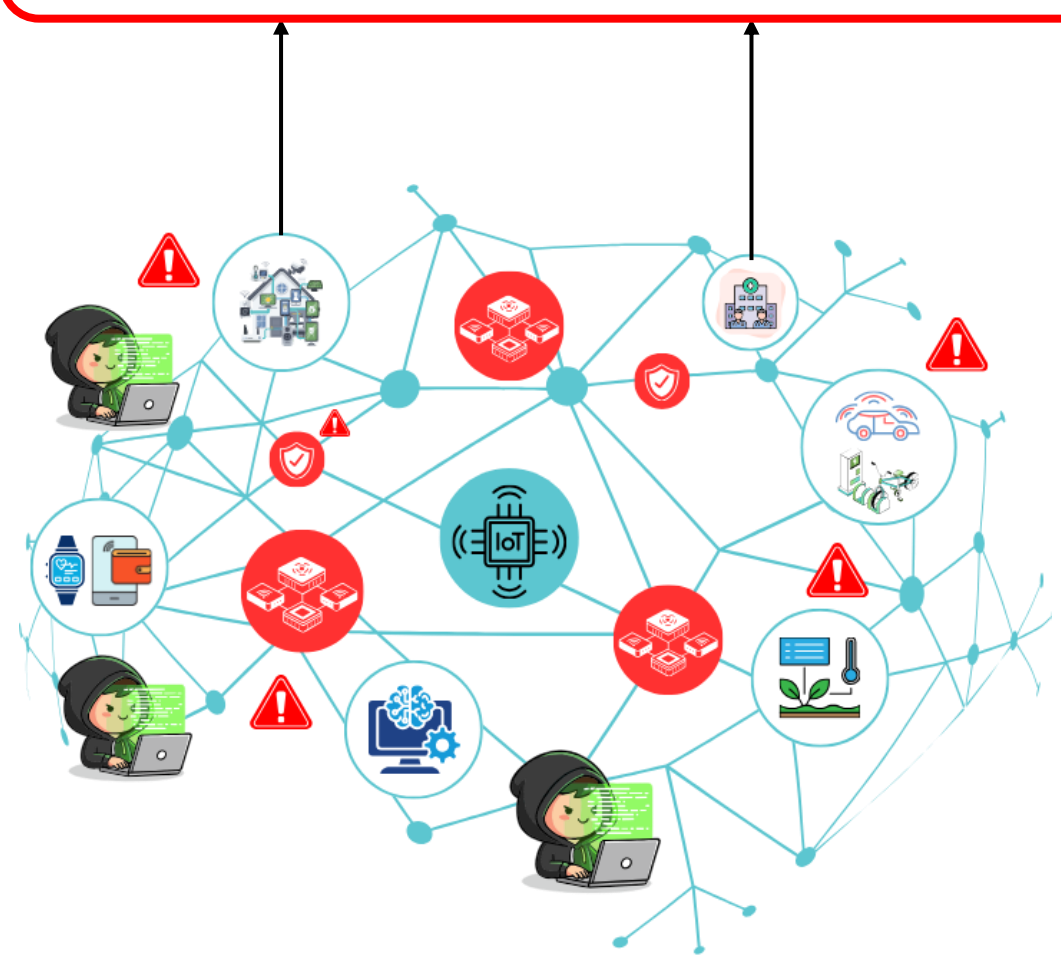


Figure: Hyper-connected IoT–edge ecosystem showing interconnected devices and domains. Red markers indicate potential attack entry points.

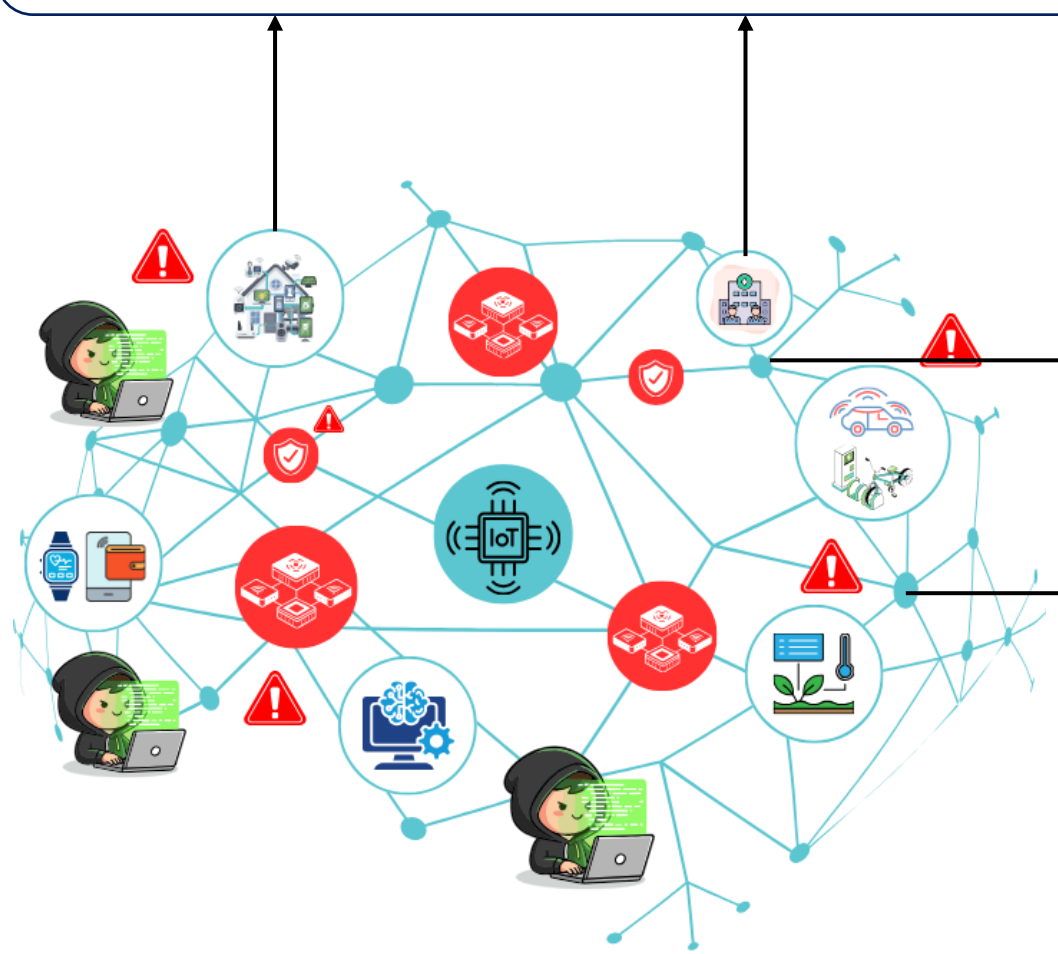
Challenges in development of Intrusion Detection Systems

Domain shift: In the depicted heterogeneous IoT network, an IDS trained on one environment (e.g., smart home traffic) may encounter different traffic patterns when deployed in other environments (e.g., vehicular or industrial IoT), causing performance degradation. Robust generalization across diverse network domains is therefore essential.

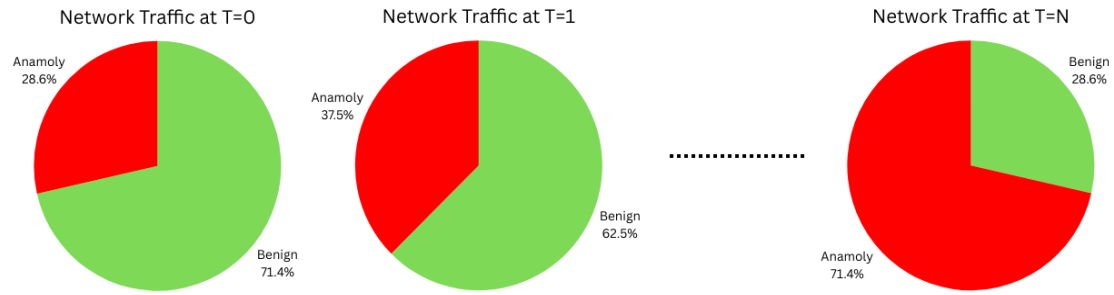


Challenges in development of Intrusion Detection Systems

Domain shift: In the depicted heterogeneous IoT network, an IDS trained on one environment (e.g., smart home traffic) may **encounter different traffic patterns** when deployed in other environments (e.g., vehicular or industrial IoT), causing performance degradation. Robust generalization across diverse network domains is therefore essential.

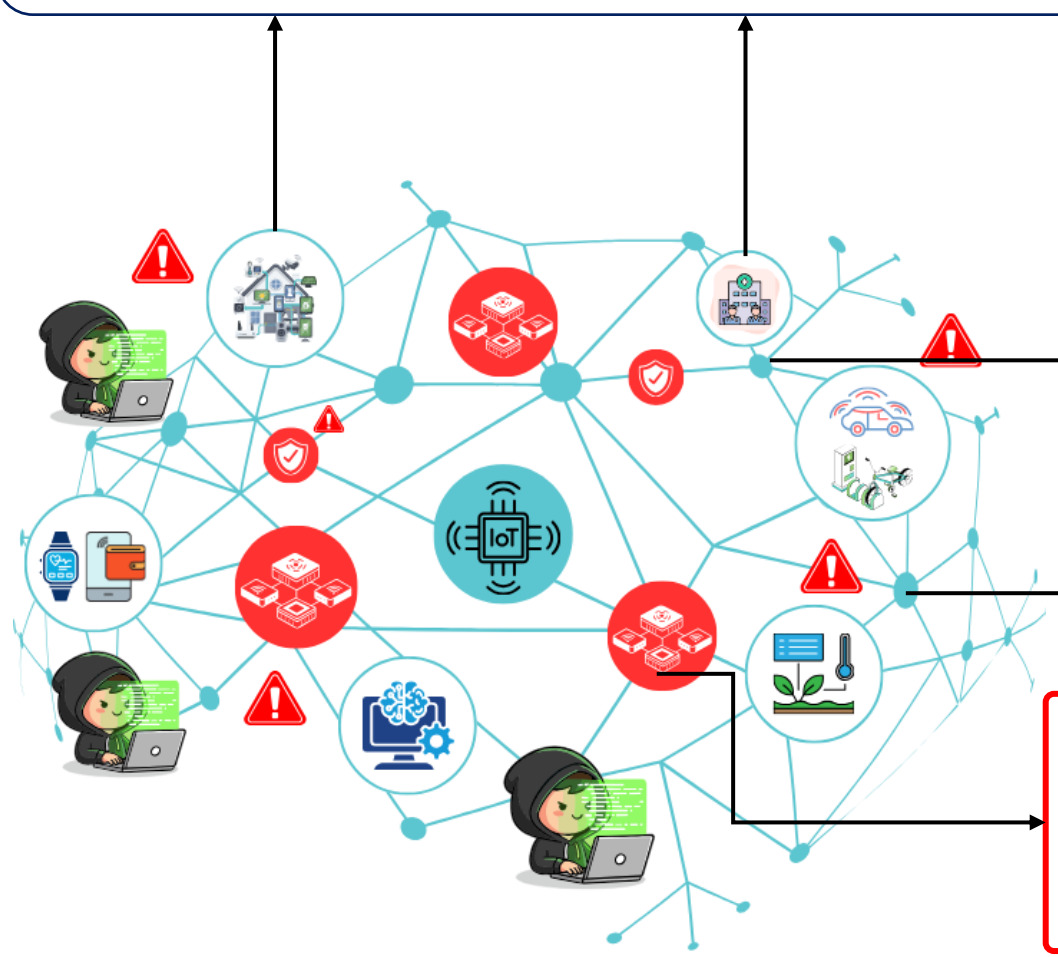


Concept drift: Over time, normal and malicious IoT traffic patterns gradually evolve, **causing previously trained IDS models to lose accuracy if not continuously updated.**

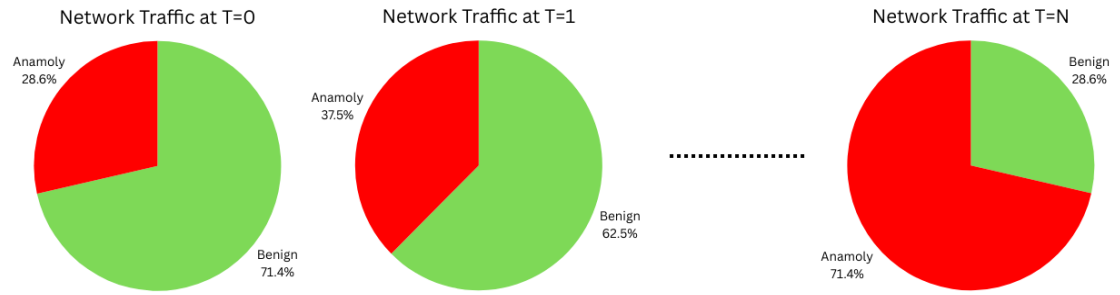


Challenges in development of Intrusion Detection Systems

Domain shift: In the depicted heterogeneous IoT network, an IDS trained on one environment (e.g., smart home traffic) may encounter different traffic patterns when deployed in other environments (e.g., vehicular or industrial IoT), causing performance degradation. Robust generalization across diverse network domains is therefore essential.



Concept drift: Over time, normal and malicious IoT traffic patterns gradually evolve, causing previously trained IDS models to lose accuracy if not continuously updated.



Concept shift: Concept shift occurs when entirely new and previously unseen attack types emerge in the IoT network, forcing IDS models trained on known classes to misclassify these **zero-day threats.**

Related Work

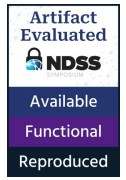
- The application of standard Machine Learning (ML) and Deep Learning (DL) models to IoT-based NIDS has been extensively studied and demonstrated high accuracy and a lower false positive rate in classifying attacks within static datasets [1-4]. **However, this classifiers are static and inherently incapable of handling concept shift and concept drift problems**.

- To overcome this static learning problem, a more adaptive paradigm is required. Recent work, such as MATEEN, has focused on building online learning frameworks to handle concept drift in benign traffic, using ensembles of autoencoders to adapt to evolving network behaviors [5]. **While effective for adapting to known classes, this approach does not explicitly address the concept shift/zero-day problem.**

- Meta-learning was adopted to handle concept shift by enabling rapid adaptation to new threats, yet its reliance on deterministic embeddings fails because network traffic is inherently **stochastic and unpredictable.** Deterministic models represent data as fixed points that cannot capture this high level of uncertainty, meaning they still struggle to **generalize and the problem of concept shift remains fundamentally unsolved [6-9].**

- Existing IoT datasets lack high-fidelity, evolving threats, and current IDS literature lacks validated testing on edge devices

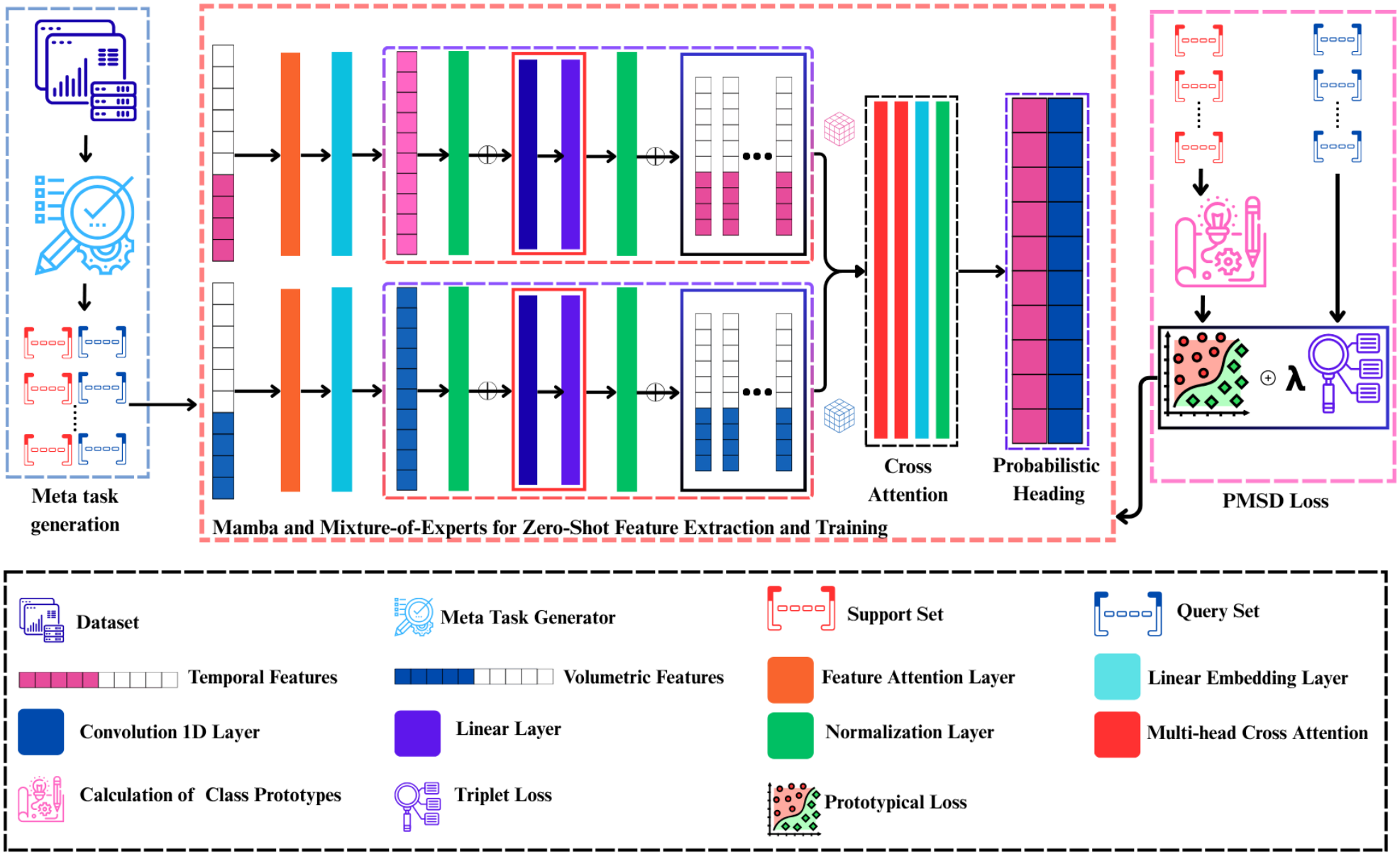
Problem Statement and Our Contributions



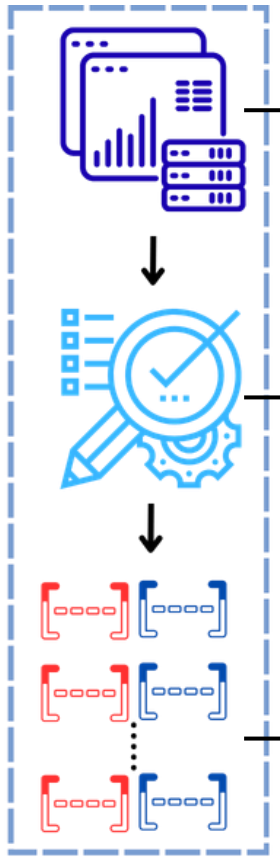
Problem Statement: Current IDS models struggle to handle domain shift, evolving traffic behaviors, and zero-day attacks in heterogeneous IoT environments, leading to performance degradation in real-world deployment. **Pandora** address this by designing an **uncertainty-aware, edge-efficient IDS framework** capable of robust generalization and zero-shot threat detection.

- Adversarial IoT Dataset: We release a high-fidelity IoT intrusion dataset with subclass-level attacks and heterogeneous traffic.
- **PANDORA Framework:** We propose an edge-efficient architecture integrating uncertainty-aware embeddings, a new PMSD loss for zero-shot generalization, and a lightweight Mamba-Mixture-of-Expert backbone.
- Adaptation and Intelligence: We enable zero-shot detection and few-shot prototype adaptation, offering contextual threat insights beyond binary alerts.
- Real-World Validation: We demonstrate end-to-end deployment on a resource-constrained IoT testbed, showing field-ready performance.
- Code and Dataset Availability: Code is available at **https://doi.org/10.5281/zenodo.17881774** .

Major Contribution: PANDORA Framework Overview



PANDORA: Zero Shot Meta Task Generation



Meta task generation

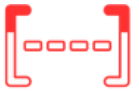
We used the **CICIDS2017** and **CICIoT2023** benchmark datasets to perform subclass-level multi-class intrusion classification. Additionally, our **TTDFIoTIDS2025** dataset was incorporated to evaluate the model's generalization across diverse IoT attack scenarios.

Meta-task generation creates many small training episodes by randomly selecting classes and splitting their samples into a **support set** (for fast adaptation) and a **query set** (for evaluation).

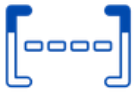
Support and query sets are further processed through parallel modality-specific blocks to extract and fuse both packet-level content and long-range temporal-window patterns.

Dataset

Meta Task Generator



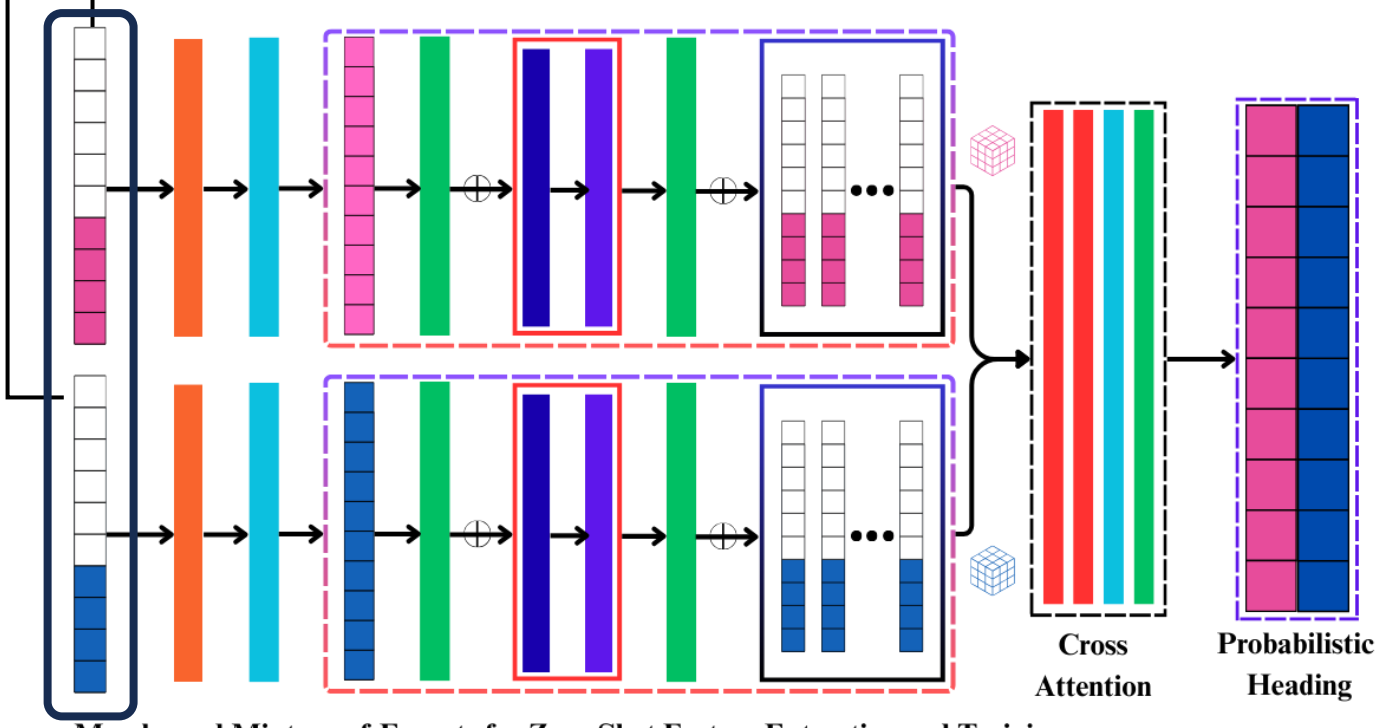
Support Set



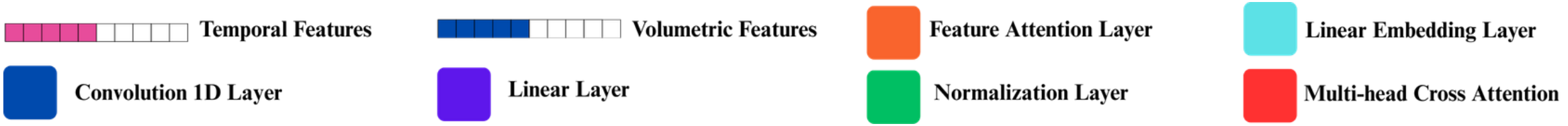
Query Set

PANDORA: Mamba-Mixture-of-Experts Backbone

Modality Splitting: Input traffic features are divided into temporal and volumetric modalities to allow modality-specific representation learning. The features are split in multiple modality to avoid feature bias and parallel training for faster training and inference.

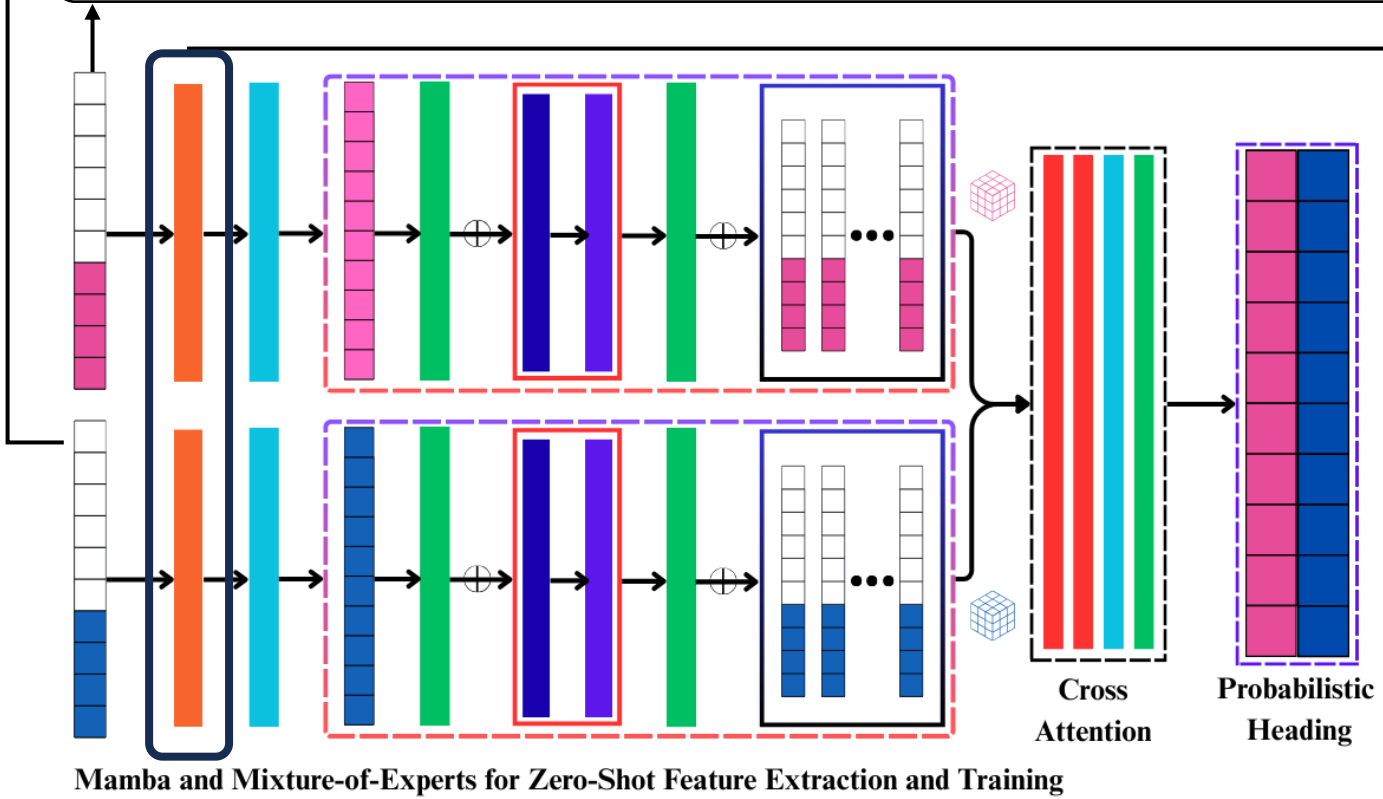


Mamba and Mixture-of-Experts for Zero-Shot Feature Extraction and Training



PANDORA: Mamba-Mixture-of-Experts Backbone

Modality Splitting: Input traffic features are divided into temporal and volumetric modalities to allow modality-specific representation learning. The features are split in multiple modality to avoid feature bias and parallel training for faster training and inference.

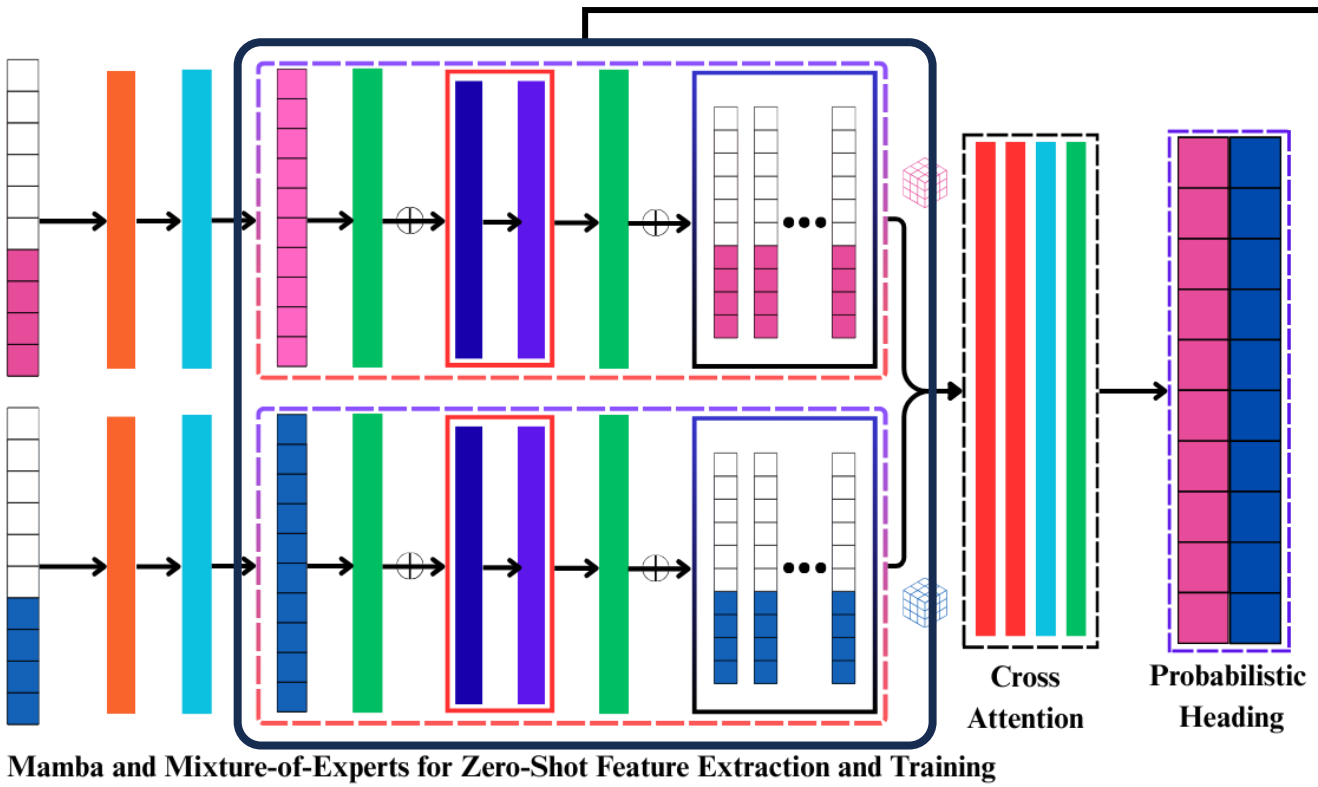


Mamba and Mixture-of-Experts for Zero-Shot Feature Extraction and Training

Feature Attention Layer:
A learnable attention mechanism computes feature-importance weights using a softmax operation and multiplies them with the input features to emphasize informative attributes.

Temporal Features
Volumetric Features
Feature Attention Layer
Linear Embedding Layer
Convolution 1D Layer
Linear Layer
Normalization Layer
Multi-head Cross Attention

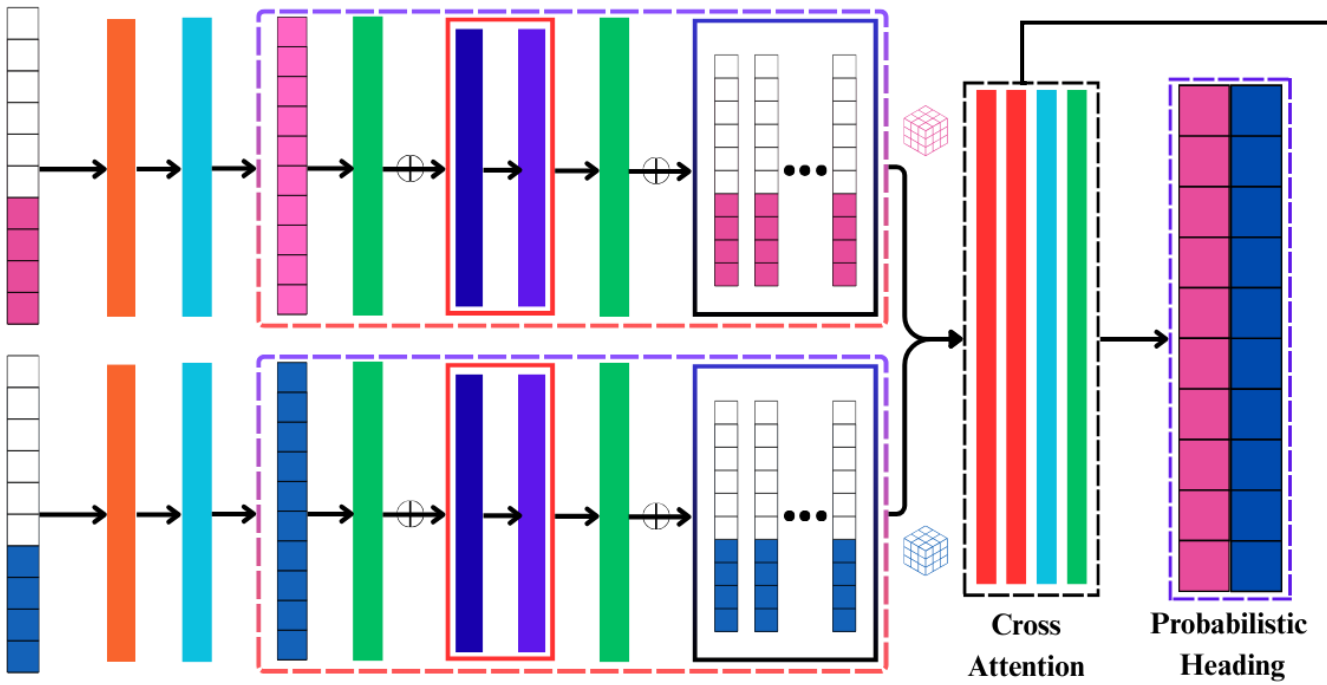
PANDORA: Mamba-Mixture-of-Experts Backbone



Mamba and Mixture-of-Experts for Zero-Shot Feature Extraction and Training

Mamba–Mixture-of-Experts Encoder:
The attended features are processed through Mamba state-space layers with a Mixture-of-Experts routing mechanism, enabling efficient long-range dependency modeling while activating only relevant experts for each modality. The Mamba backbone provides **linear-time complexity**, lower than Transformer-based models, and expert routing reduces effective parameter usage, resulting in **faster and more efficient inference**.

PANDORA: Mamba-Mixture-of-Experts Backbone



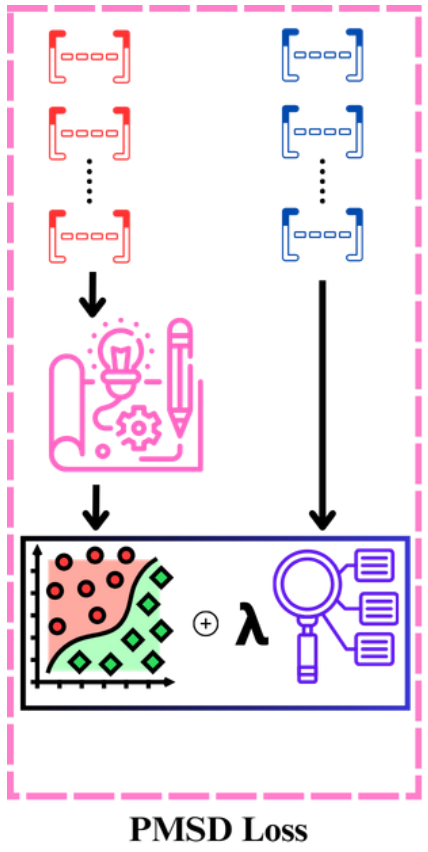
Mamba and Mixture-of-Experts for Zero-Shot Feature Extraction and Training

Cross-Attention Fusion Module
Temporal and volumetric embeddings attend to each other through multi-head cross-attention, producing a unified fused representation capturing complementary information.

Probabilistic Embedding Head
Two parallel linear projections generate the **mean and variance vectors** of Gaussian embeddings, allowing the model to represent feature uncertainty instead of deterministic points.

Temporal Features
Volumetric Features
Feature Attention Layer
Linear Embedding Layer
Convolution 1D Layer
Linear Layer
Normalization Layer
Multi-head Cross Attention

The Probabilistic Manifold Structuring and Distance (PMSD) Loss: **Adaptive Metric Learning**



$$\mathcal{L}_{\text{Wass}} = -\log \frac{\exp(-d(f_\phi(\boldsymbol{x}_j), \boldsymbol{c}_{y_j}))}{\sum_{k=1}^{N} \exp(-d(f_\phi(\boldsymbol{x}_j), \boldsymbol{c}_k))}$$

$$\mathcal{L}_{\text{Triplet}} = \max(\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_p\|_2^2 - \|\boldsymbol{\mu}_a - \boldsymbol{\mu}_n\|_2^2 + m, 0)$$

$$\mathcal{L}_{\text{PMSD}} = \mathcal{L}_{\text{Wass}} + \lambda \cdot \mathcal{L}_{\text{Triplet}}$$

$$\mathcal{L}_{PMSD} = \frac{1}{2\sigma_{Wass}^2}\mathcal{L}_{Wass} + \frac{1}{2\sigma_{Triplet}^2}\mathcal{L}_{Triplet} + \log(\sigma_{Wass}) + \log(\sigma_{Triplet})$$

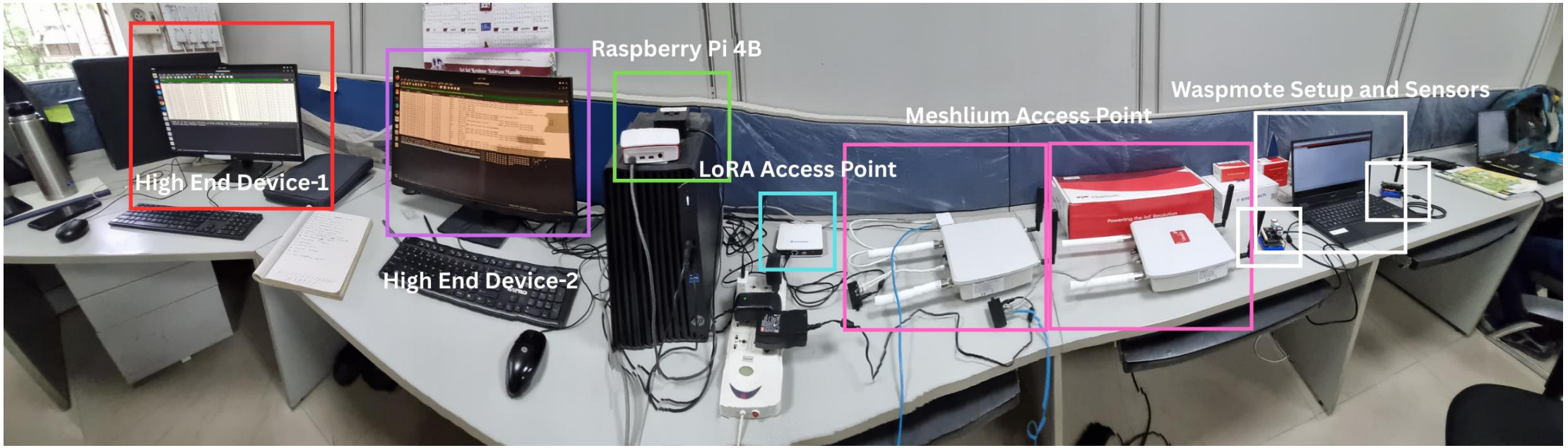
Auto balance Loss based on Homoscedastic Uncertainty [10]

The Wasserstein loss encourages the embedding of a query sample to be closer to its correct class prototype than to other class prototypes. Here, the encoder generates the sample embedding, class prototypes represent each class, and the distance function measures similarity across all classes.

The triplet loss ensures that the embedding of an anchor sample is closer to its positive (same-class) embedding than to a negative (different-class) embedding by at least a margin, enforcing class separation in the feature space.

PMSD enables adaptiveness by dynamically adjusting loss weighting based on data uncertainty, allowing flexible decision boundaries.

TTDFIoTIDS2025 - Modern Testbed with Adaptive Attacks



Realistic Heterogeneous Testbed: The dataset was generated in a physical IoT environment with Raspberry Pi edge nodes, Wi-Fi access points, mesh, and LoRaWAN networks, forming a **hybrid (star–mesh–LPWAN) heterogeneous IoT topology.**
Attack Generation: High-end systems generated external and internal attacks and collected packets centrally.
Data Collection: Traffic was captured as PCAP files using Wireshark, and CICFlowMeter + dpkt extracted 83 flow and packet-level features.

TTDFIoTIDS2025 - Modern Testbed with Adaptive Attacks

- **Enhanced Realism:** Unlike the simplistic, centralized topology of testbeds like CICIoT2023, CICIDS2017, our setup uses a heterogeneous wireless infrastructure (multiple APs, LoRa, mesh) that more accurately reflects complex, real-world IoT environments.
- **Advanced Attack Generation:** We move beyond the predictable, tool-based attacks found in datasets like CI CIDS2017 by employing custom, programmatic methods that better mimic sophisticated adversarial behavior.
- **Complex Attack Scenarios:** In contrast to the isolated attacks or tool-based attacks seen in other datasets, our dataset has features of concurrent, multi-threaded attacks from multiple sources to simulate modern threat campaigns.
- **Greater Threat Granularity:** While existing datasets have broad attack categories, seven in CICIoT2023 and 33 at the sub-class level, our dataset provides a richer taxonomy with nine classes and **63 sub-classes**, enabling a more fine-grained evaluation of detection models.

Statistic	Full Dataset	Evaluation Subset
Total Flows	4,742,843	20,00,000
Benign Flows	183,179	50,000
Malicious Flows	4,559,664	15,00,000
Number of Features	83	66
Major Attack Categories	9	7
Attack Sub-classes	63	30

Full Dataset: Enables **large-scale IDS model training, detailed attack sub-class analysis, feature learning, and community-wide benchmarking** due to its comprehensive size and diversity.

TTDFIoTIDS2025 - Modern Testbed with Adaptive Attacks

- **Enhanced Realism:** Unlike the simplistic, centralized topology of testbeds like CICIoT2023, CICIDS2017, our setup uses a heterogeneous wireless infrastructure (multiple APs, LoRa, mesh) that more accurately reflects complex, real-world IoT environments.
- **Advanced Attack Generation:** We move beyond the predictable, tool-based attacks found in datasets like CI CIDS2017 by employing custom, programmatic methods that better mimic sophisticated adversarial behavior.
- **Complex Attack Scenarios:** In contrast to the isolated attacks or tool-based attacks seen in other datasets, our dataset has features of concurrent, multi-threaded attacks from multiple sources to simulate modern threat campaigns.
- **Greater Threat Granularity:** While existing datasets have broad attack categories, seven in CICIoT2023 and 33 at the sub-class level, our dataset provides a richer taxonomy with nine classes and **63 sub-classes**, enabling a more fine-grained evaluation of detection models.

Statistic	Full Dataset	Evaluation Subset
Total Flows	4,742,843	20,00,000
Benign Flows	183,179	50,000
Malicious Flows	4,559,664	15,00,000
Number of Features	83	66
Major Attack Categories	9	7
Attack Sub-classes	63	30

Evaluation Subset: Supports standardized performance evaluation, faster experimentation, reproducible benchmarking, and fair comparison of different detection methods. Malicious Flows contains sub class level flows all in total for attack sub classes.

Experimental Setup for Results and Analysis

- **Evaluation Objectives:** Experiments evaluate whether **TTDFIoTIDS2025** provides a more realistic benchmark, measure **PANDORA's performance on known and zero-day attacks**, justify its architectural components, and assess **real-time deployability in resource-constrained environments**.
- **Hardware Setup:** Training was conducted on a **RTX 3070 GPU workstation (64GB RAM)**, while **deployment testing** was performed on a **Raspberry Pi 4B (8GB RAM)**.
- **Baseline and Fair Comparison:** A **re-implemented PTN-IDS transformer-based baseline** was used to ensure **fair performance comparison**, isolating the algorithmic improvements of PANDORA with **lightweight hyperparameters suitable for edge deployment**. **Dataset such as CICIDS2017, CICIDS2018 and CICIoT2023 used for comparison.**

Architectural Parameters		Training Parameters	
Parameter	**Value**	**Parameter**	**Value**
d_{model} (Embedding Dim)	64	Optimizer	Adam
num_blocks (Mamba-MoE)	1	Learning Rate	1×10^{-4}
$num\_experts$ (per MoE)	2	Triplet Margin (m)	1.0
n_{heads} (Fusion Attention)	2	Batch Size	256
$dropout\_rate$	0.5		

Architectural Parameters: The hyperparameter values were determined through extensive ablation studies across different value ranges. The results showed only negligible performance variation around the selected values; therefore, the **chosen configuration provides a stable and reliable setting for all experiments**.

Results and Analysis: SOTA Concept Shift/Drift Comparison

SOTA Comparison on CICIDS2017 (Few-Shot Adaptation, Acc = Accuracy, F1 = F1-score)

Shots	Metric	Sc.1 (n=1)		Sc.2 (n=2)		Sc.3 (n=3)	
		PTN	**PANDORA**	PTN	**PANDORA**	PTN	**PANDORA**
1-shot	Acc	0.792	**0.890**	0.701	0.680	0.635	0.576
	F1	0.765	**0.914**	0.680	0.586	0.592	0.554
5-shot	Acc	0.910	**0.961**	0.830	**0.897**	0.795	**0.835**
	F1	0.907	**0.967**	0.823	**0.879**	0.779	**0.831**
10-shot	Acc	0.931	**0.969**	0.845	**0.908**	0.819	**0.860**
	F1	0.930	**0.981**	0.837	**0.894**	0.808	**0.879**

Comparison of Distance Functions in PTN-IDS vs PANDORA (5-shot, CICIDS2017)

Dist.	Sc.1 (n=1)		Sc.2 (n=2)		Sc.3 (n=3)	
	Acc	F1	Acc	F1	Acc	F1
Euclidean	0.910	0.907	0.830	0.823	0.795	0.779
Manhattan	0.886	0.878	0.813	0.804	0.780	0.768
Cosine	0.910	0.905	0.729	0.696	0.769	0.756
Wasserstein	**0.961**	**0.967**	**0.897**	**0.879**	**0.841**	**0.821**

Scenario Representation: The evaluation uses three scenarios representing **increasing adaptation difficulty**: **Sc.1 (n=1)** – single unseen attack class (simplest), **Sc.2 (n=2)** – two unseen heterogeneous classes (moderate), and **Sc.3 (n=3)** – multiple unseen classes (most complex), reflecting realistic few-shot intrusion detection challenges.

Few-Shot Adaptation Performance: PANDORA consistently outperforms PTN-IDS in **5-shot and 10-shot settings**, achieving the highest F1-scores, demonstrating stronger adaptation capability, especially as more target samples become available.

Wasserstein (PMSD) Ablation: The Wasserstein-based PMSD loss outperforms Euclidean, Manhattan, and Cosine metrics, improving accuracy and enabling strong cross-domain generalization (~0.99 accuracy without finetuning).

Results and Analysis: SOTA Domain Shift Comparison

PERFORMANCE UNDER DOMAIN SHIFT (TRAIN: CICIDS2017, TEST: CICIDS2018)

Method	Ben	DDoS	BF	Bot	Web	DoS	OA
Baseline	0.995	0.040	0.000	0.000	0.000	0.000	0.172
PTN-IDS 5-shot	0.707	0.870	0.540	0.569	0.765	0.555	0.668
+ Fine-Tuning	0.826	1.000	0.994	0.959	0.981	0.945	0.951
PANDORA 1-shot	0.225	0.254	1.000	0.872	0.310	0.867	0.565
PANDORA 5-shot	0.321	0.424	1.000	0.864	0.512	0.838	0.669
+Zero Shot	**0.987**	**1.000**	**1.000**	**1.000**	**0.992**	**0.989**	**0.991**

Robustness and Cross-Domain Generalization: Using the PMSD (Wasserstein-based) distance improves performance over standard metrics, and **zero-shot cross-domain testing (CICIDS2017 → CICIDS2018)** shows strong generalization, achieving **~0.99 accuracy without finetuning**, indicating robustness to domain shifts.

QUALITATIVE AND QUANTITATIVE SOTA COMPARISON OF NIDS FRAMEWORK CAPABILITIES.

Model	Best F1-Score	Zero-Shot	Few-Shot Adaptation	Edge Deploy.	Complex Dataset	Feat. Attn.	Dataset(s) Used	Classes Eval.
RF	0.7823	✗	✗	✓	✗	✗	Various	Various
MAML-NIDS [26]	0.9219	✗	✓	✗	✗	✗	CICIDS2017	5 Major
MAML-NIDS Online [55]	0.9285	✗	✓	✗	✗	✗	CICIDS2017	5 Major
Transformer-NIDS [61]	0.9310	✗	✗	✗	✗	✗	NSL-KDD	5 Major
PTN-IDS [58]	0.9310	✗	✓	✗	✗	✗	CICIDS2017	7 Major
MASiNet [57]	0.9412	✗	✓	✗	✗	✗	UNSW-NB15, NSL-KDD	10 Major
Helios [59]	0.9578	✗	✗	✗	✗	✗	CICIoT2023 TON-IoT	12 Sub-classes
MTH-IDS [80]	0.9630	✗	✗	✗	✗	✗	CICIDS2017	20 Sub-classes
PANDORA Zero Shot (Ours)	**0.7982,0.8983,0.9690**	✓	✓	✓	✓	✓	**TTDF, CICIoT2023, CICIDS2017**	**41,23,7 Sub Classes**
PANDORA Known (Ours)	**0.8203, 0.9182, and 0.9983**	✓	✓	✓	✓	✓	**TTDF, CICIoT2023, CICIDS2017**	**41,23,7 Sub Classes**

Background for Real-Time On-Device Performance: Zero-Shot Detection and Few-Shot Adaptation

Zero-Shot Detection: For each incoming network flow, the model generates an embedding and measures its distance to all known class prototypes. If the **minimum distance (novelty score)** exceeds an **adaptive threshold computed from validation-distance statistics**, the sample is flagged as a **potential zero-day attack**, while the closest prototype label is suggested as contextual guidance.

$$\text{If } \text{NoveltyScore} = \min_k d(f_\phi(\boldsymbol{x}_{\text{new}}), \boldsymbol{c}_k)$$

$$\vee$$

$$\tau_{\text{soft}} = Q_{0.95}(\mathcal{D}_{\text{val}}) + \epsilon$$

$$=$$

$$\text{SuggestedLabel} = k^* = \arg\min_k d(f_\phi(\boldsymbol{x}_{\text{new}}), \boldsymbol{c}_k)$$

Background for Real-Time On-Device Performance: Zero-Shot Detection and Few-Shot Adaptation

Zero-Shot Detection: For each incoming network flow, the model generates an embedding and measures its distance to all known class prototypes. If the **minimum distance (novelty score)** exceeds an **adaptive threshold computed from validation-distance statistics**, the sample is flagged as a **potential zero-day attack**, while the closest prototype label is suggested as contextual guidance.

$$\text{If NoveltyScore} = \min_k d(f_\phi(\boldsymbol{x}_{\text{new}}), \boldsymbol{c}_k)$$

$$\vee$$

$$\tau_{\text{soft}} = Q_{0.95}(\mathcal{D}_{\text{val}}) + \epsilon$$

$$=$$

$$\text{SuggestedLabel} = k^* = \arg\min_k d(f_\phi(\boldsymbol{x}_{\text{new}}), \boldsymbol{c}_k)$$

Few-Shot Adaptation: When a few labeled samples of the detected new attack become available, the framework performs **incremental fine-tuning**, quickly incorporating the new class into the model to improve future detection accuracy.

Input: Trained Model f_ϕ, Training Data D_{train}, Adaptation Set D_{adapt}, Episodes E_{adapt}, LR η'
Output: Fine-tuned Model $f_{\phi'}$
$\phi' \leftarrow \phi$
$D_{\text{combined}} \leftarrow D_{\text{train}} \cup D_{\text{adapt}}$
for $e = 1$ to E_{adapt} **do**
    Sample support set S and query set Q from D_{combined}
    Compute $\mathcal{L}_{\text{PMSD}}$ using Algorithm 1
    $\phi' \leftarrow \phi' - \eta' \nabla_{\phi'} \mathcal{L}_{\text{PMSD}}$
end for
return $f_{\phi'}$

Results and Analysis: Real-Time Device Deployment of PANDORA

ON-DEVICE PERFORMANCE: BASELINE / **PANDORA**. **BOLD** DENOTES MAMBA-MOE RESULTS.

Metric	**CICIDS2017** (Baseline / **Ours**)	**CICIoT2023** (Baseline / **Ours**)	**TTDF-IoT-2025** (Baseline / **Ours**)
Throughput (Flows/sec)	0.52 / **2.03**	1.12 / **4.26**	0.98 / **3.82**
Latency (ms/flow)	215.10 / **58.20**	142.50 / **37.34**	135.20 / **34.76**
CPU Usage (%)	345.20 / **118.70**	355.80 / **125.60**	340.50 / **120.20**
Memory Usage (MB)	85.60 / **24.48**	82.40 / **24.16**	84.10 / **24.30**

Edge Efficiency: PANDORA achieves **3–4× higher throughput with lower latency** while maintaining a **lightweight resource footprint (~24MB memory, ~1.2 CPU cores)**, significantly outperforming baseline models that consume higher CPU and memory, making them less suitable for edge IoT deployment.

Results and Analysis: Real-Time Device Deployment of PANDORA

ON-DEVICE PERFORMANCE: BASELINE / **PANDORA**. **BOLD** DENOTES MAMBA-MOE RESULTS.

Metric	CICIDS2017 (Baseline / **Ours**)	CICIoT2023 (Baseline / **Ours**)	TTDF-IoT-2025 (Baseline / **Ours**)
Throughput (Flows/sec)	0.52 / **2.03**	1.12 / **4.26**	0.98 / **3.82**
Latency (ms/flow)	215.10 / **58.20**	142.50 / **37.34**	135.20 / **34.76**
CPU Usage (%)	345.20 / **118.70**	355.80 / **125.60**	340.50 / **120.20**
Memory Usage (MB)	85.60 / **24.48**	82.40 / **24.16**	84.10 / **24.30**

Edge Efficiency: PANDORA achieves **3–4× higher throughput with lower latency** while maintaining a **lightweight resource footprint (~24MB memory, ~1.2 CPU cores)**, significantly outperforming baseline models that consume higher CPU and memory, making them less suitable for edge IoT deployment.

ABLATION OF ENCODER ARCHITECTURE ACROSS DATASETS

Dataset	Config	Params	Time (ms)	F1	AUC
CIC17	**Mamba-MoE**	196,938	**0.0179**	**0.836**	**0.991**
	Transformer	183,552	0.1595	0.509	0.927
CICIoT23	**Mamba-MoE**	195,378	**0.0181**	**0.813**	**0.985**
	Transformer	183,552	0.1165	0.356	0.919
TTDF25	**Mamba-MoE**	195,378	**0.0176**	**0.813**	**0.985**
	Transformer	183,552	0.1669	0.356	0.919

Encoder Ablation: In minimum-epoch ablation experiments, the **Mamba-MoE encoder** achieves **much lower inference time (~0.017–0.018 ms)** and **higher F1/AUC** than the Transformer. Although parameters are slightly higher, **MoE modality-based gating activates only relevant experts**, enabling **faster and more efficient inference**.

Results and Analysis: Real-Time Device Deployment of PANDORA

ON-DEVICE PERFORMANCE: BASELINE / **PANDORA**. **BOLD** DENOTES MAMBA-MOE RESULTS.

Metric	CICIDS2017 (Baseline / **Ours**)	CICIoT2023 (Baseline / **Ours**)	TTDF-IoT-2025 (Baseline / **Ours**)
Throughput (Flows/sec)	0.52 / **2.03**	1.12 / **4.26**	0.98 / **3.82**
Latency (ms/flow)	215.10 / **58.20**	142.50 / **37.34**	135.20 / **34.76**
CPU Usage (%)	345.20 / **118.70**	355.80 / **125.60**	340.50 / **120.20**
Memory Usage (MB)	85.60 / **24.48**	82.40 / **24.16**	84.10 / **24.30**

Edge Efficiency: PANDORA achieves **3–4× higher throughput with lower latency** while maintaining a **lightweight resource footprint (~24MB memory, ~1.2 CPU cores)**, significantly outperforming baseline models that consume higher CPU and memory, making them less suitable for edge IoT deployment.

ABLATION OF ENCODER ARCHITECTURE ACROSS DATASETS

Dataset	Config	Params	Time (ms)	F1	AUC
CIC17	**Mamba-MoE**	196,938	**0.0179**	**0.836**	**0.991**
	Transformer	183,552	0.1595	0.509	0.927
CICIoT23	**Mamba-MoE**	195,378	**0.0181**	**0.813**	**0.985**
	Transformer	183,552	0.1165	0.356	0.919
TTDF25	**Mamba-MoE**	195,378	**0.0176**	**0.813**	**0.985**
	Transformer	183,552	0.1669	0.356	0.919

Encoder Ablation: In minimum-epoch ablation experiments, the **Mamba-MoE encoder** achieves **much lower inference time (~0.017–0.018 ms)** and **higher F1/AUC** than the Transformer. Although parameters are slightly higher, **MoE modality-based gating activates only relevant experts**, enabling **faster and more efficient inference**.

Zero-Day Detection: The novelty-score method achieved **98–100% detection of unseen zero-day attacks** across all datasets on the edge testbed.

Discussions and Future Work

Shot	AUC (100:1)	AUC (1000:1)	AUC (10000:1)	Acc (100:1)	Acc (1000:1)	Acc (10000:1)
1-Shot	0.95	0.97	0.97	0.94	0.93	0.93
5-Shot	0.97	0.97	0.97	0.99	0.99	0.99
10-Shot	0.97	0.97	0.96	0.98	0.97	0.97

(1) Real IoT networks exhibit extreme imbalance with benign to-attack ratios such as 100:1, 1000:1 and 10,000:1. PANDORA mitigates this by using episodic meta-learning, which constructs balanced N-way, K-shot episodes irrespective of raw distribution. As validated in Table, prototype learning is driven by feature modality rather than frequency, allowing PANDORA to sustain stable performance under increasing imbalance.

(2) Horizontal Scalability: Due to its **lightweight footprint (~24 MB)**, PANDORA supports **distributed edge deployment**, maintaining **constant per-node processing (O(1))** and stable detection latency (~37 ms) even when scaling to **1000+ IoT nodes**.

(3) Vertical Efficiency: The **Mamba-MoE backbone provides linear complexity (O(L)) and O(1) inference**, enabling real-time processing under high traffic loads, whereas transformer-based NIDS scale quadratically ($O(L^2)$), leading to rapid computational saturation.

Discussions and Future Work

Shot	AUC (100:1)	AUC (1000:1)	AUC (10000:1)	Acc (100:1)	Acc (1000:1)	Acc (10000:1)
1-Shot	0.95	0.97	0.97	0.94	0.93	0.93
5-Shot	0.97	0.97	0.97	0.99	0.99	0.99
10-Shot	0.97	0.97	0.96	0.98	0.97	0.97

(1) Real IoT networks exhibit extreme imbalance with benign to-attack ratios such as 100:1, 1000:1 and 10,000:1. PANDORA mitigates this by using episodic meta-learning, which constructs balanced N-way, K-shot episodes irrespective of raw distribution. As validated in Table, prototype learning is driven by feature modality rather than frequency, allowing PANDORA to sustain stable performance under increasing imbalance.

(2) Horizontal Scalability: Due to its **lightweight footprint (~24 MB)**, PANDORA supports **distributed edge deployment**, maintaining **constant per-node processing (O(1))** and stable detection latency (~37 ms) even when scaling to **1000+ IoT nodes**.

(3) Vertical Efficiency: The **Mamba-MoE backbone provides linear complexity (O(L)) and O(1) inference**, enabling real-time processing under high traffic loads, whereas transformer-based NIDS scale quadratically ($O(L^2)$), leading to rapid computational saturation.

Future Work: Future research will focus on improving robustness under **extreme class-imbalance scenarios with higher attack–benign ratios** and extending the framework's **scalability to ultra-large IoT deployments**, including large-scale distributed edge clusters and high-density traffic environments.

Thank You! Scan to Access Lab, Dataset & Authors

Avinash Awasthi

Pritam Vediya

Hemant Miranka

Ramesh B Battula

Manoj S Gaur

FICS LAB

Code and Dataset

References

[1] F. Wei, H. Li, Z. Zhao, and H. Hu, “xNIDS: Explaining deep learning-based network intrusion detection systems for active intrusion responses,” in 32nd USENIX Security Symposium (USENIX Security 23). Anaheim, CA: USENIX Association, Aug. 2023, pp. 4337–4354. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/wei-feng

[2] D. K. Nkashama, A. Soltani, J.-C. Verdier, M. Frappier, P.-M. Tardif, and F. Kabanza, “Robustness evaluation of deep unsupervised learning algorithms for intrusion detection systems,” 2023. [Online]. Available: https://arxiv.org/abs/2207.03576

[3] L. Yang and A. Shami, “Towards autonomous cybersecurity: An intelli gent automl framework for autonomous intrusion detection,” in Proceed ings of the Workshop on Autonomous Cybersecurity (AutonomousCyber ’24), ACM Conference on Computer and Communications Security (CCS) 2024, Salt Lake City, UT, USA, 2024, pp. 1–11.

[4] I. Mohammed Sayem, M. Islam Sayed, S. Saha, and A. Haque, “Enids: A deep learning-based ensemble framework for network intrusion detection systems,” IEEE Transactions on Network and Service Management, vol. 21, no. 5, pp. 5809–5825, 2024.

[5] F. Alotaibi and S. Maffeis, “Mateen: Adaptive ensemble learning for network anomaly detection,” in Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses, ser. RAID ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 215–234. [Online]. Available: https://doi.org/10.1145/3678890.3678901

References

[6] Z. Wang, Y. Lu, W. Wu, Y. Lu, and H. Wang, “A few-shot and antiforgetting network intrusion detection system based on online meta learning,” in GLOBECOM 2024- 2024 IEEE Global Communications Conference, 2024, pp. 3877–3882.

[7] U. Zukaib, X. Cui, C. Zheng, M. Hassan, and Z. Shen, “Meta-ids: Meta learning-based smart intrusion detection system for internet of medical things (iomt) network,” IEEE Internet of Things Journal, vol. 11, no. 13, pp. 23080–23095, 2024.

[8] Y. Wu, G. Lin, L. Liu, Z. Hong, Y. Wang, X. Yang, Z. L. Jiang, S. Ji, and Z. Wen, “Masinet: Network intrusion detection for iot security based on meta-learning framework,” IEEE Internet of Things Journal, vol. 11, no. 14, pp. 25136–25146, 2024.

[9] N. Niknami, V. Mahzoon, and J. Wu, “Ptn-ids: Prototypical network solution for the few-shot detection in intrusion detection systems,” in 2024 IEEE 49th Conference on Local Computer Networks (LCN), 2024, pp. 1–9.

[10] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” 2018.[Online]. Available: https://arxiv.org/abs/1705.07115