

Research Article

Efficient AI Architectures and Advanced Network Technologies for Future Intelligent Systems: The ICANOS Framework

Zubair Hussain* 

Independent Scholar, Doha, State of Qatar

Abstract

Transformer architectures have redefined the state of the art across natural language processing, computer vision, and signal processing, yet their quadratic computational complexity with respect to sequence length creates serious scalability and sustainability bottlenecks. At the same time, the convergence of sixth-generation (6G) wireless communication, the Internet of Things (IoT), and AI-native network intelligence introduces new complexity in resource management, proactive security, and energy governance. This paper argues that optimizing AI models or network infrastructure in isolation is insufficient: a tightly co-designed framework is needed that treats both concerns as inseparable. We survey efficient Transformer variants and hardware-aware approaches, and we characterize the networking landscape through 6G ubiquitous connectivity, AI-driven network automation, edge computing for IoT, and flexible network architectures. Drawing on these two bodies of literature, we propose the Intelligent Converged AI-Network Orchestration and Security (ICANOS) Framework, a three-layer AI-driven architecture comprising a Holistic Resource Orchestrator (HRO), a Proactive Security Manager (PSM), and an Energy Efficiency Optimizer (EEO). Unlike conventional software-defined networking controllers and network-functions-virtualization orchestrators, which treat AI models as opaque, schedulable workloads, ICANOS treats AI model behavior itself as a first-class signal for resource allocation and threat detection. Simulated benchmarks demonstrate energy savings of 57.9%, latency reduction of 63%, and a threat-detection improvement of 32.7% relative to non-integrated baselines. All experimental components use open-source, locally deployable models and publicly available datasets, and fully instrumented code with computational cost logging is provided for reproducibility.

Keywords

Efficient Transformers, 6G Networks, ICANOS, AI-Native Networking, Energy-Efficient AI, IoT Security, Federated Learning, Resource Orchestration

1. Introduction

The trajectory of modern intelligent systems is shaped by two forces pulling in tandem: the relentless growth of deep

learning model scale and the sweeping transformation of communication networks toward 6G. Transformer architectures, introduced by Vaswani et al. [1], have become the workhorse

*Correspondence: Zubair Hussain (zubairhussainwarsi@gmail.com)

Received: 14 July 2026; Accepted: 24 July 2026; Published: 30 September 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

of AI across domains, yet their self-attention mechanism incurs $O(L^2)$ time and memory complexity with sequence length L . For a sequence of 16,000 tokens this amounts to more than 256 million attention pairs per layer — a computational load that is simply unsustainable at the edge.

On the networking side, 6G promises terabit-per-second throughputs, sub-millisecond latency, and seamless integration of terrestrial, aerial, and orbital infrastructure [10, 20]. The proliferation of IoT devices is projected to exceed 100 billion endpoints globally by 2030, each generating continuous data streams that demand real-time AI inference at or near the source [12]. Managing this combination of ultra-dense connectivity, heterogeneous hardware, and dynamic AI workloads strains every existing network paradigm.

What is conspicuously absent from the literature is a holistic framework that co-designs AI model efficiency with network-level orchestration. Work on efficient Transformers tends to evaluate models on GPU clusters decoupled from realistic network constraints [2, 4, 5]. Work on next-generation networking tends to treat AI as a workload to be scheduled rather than a system component to be co-optimized [11, 19].

Existing orchestration technology — software-defined networking (SDN) controllers, network functions virtualization (NFV) management and orchestration (MANO) systems, and intent-based networking — manages compute, bandwidth, and storage as generic resources, and treats any AI model as an opaque service consuming those resources. None of these systems reasons about which model variant is running, how its internal behavior is changing, or whether its outputs can themselves be evidence of compromise. This is the specific, narrow gap that ICANOS is designed to close: it is not another orchestration framework in the abstract, but the first, to our knowledge, to make AI-model selection, AI-model integrity, and network-resource allocation a single joint decision rather than three separate ones.

This paper closes that gap with four contributions: (i) a unified survey linking efficient AI architectures and advanced network technologies through shared metrics; (ii) a formal multi-objective problem statement; (iii) the ICANOS framework with concrete algorithmic descriptions; and (iv) fully instrumented open-source code with reproducible benchmarks referencing public datasets.

2. Literature Review

2.1. Efficient Transformer Architectures

The original Transformer [1] computes attention as $\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k}) \cdot V$, where $Q, K, V \in \mathbb{R}^{L \times d}$. The pairwise dot-product step requires $O(L^2d)$ operations and $O(L^2)$ memory, making very long sequences prohibitive. Peng et al. [3] formalize several theoretical limitations, motivating the four categories below.

2.1.1. Sparse Attention Mechanisms

Reformer [2] replaces full attention with Locality-Sensitive Hashing (LSH) attention, bucketing queries into g groups and attending only within buckets, reducing complexity to $O(L \log L)$. Informer [5] introduces ProbSparse self-attention, selecting the top- u queries whose attention entropy exceeds a threshold, also achieving $O(L \log L)$ while excelling on long time-series tasks. SPARSEK Attention [6] extends this by retaining exactly k dominant keys per query through a learnable top- k selection, preserving global expressiveness while keeping memory sub-quadratic.

2.1.2. Memory-Efficient Algorithms

Reformer employs reversible residual layers that reconstruct intermediate activations during the backward pass, eliminating the need to store them explicitly [2]. Flash Attention [4] uses block tiling to reduce high-bandwidth memory (HBM) reads and writes from $O(L^2)$ to $O(L)$, while computing exact attention — delivering $2\text{--}4\times$ wall-clock speedup on A100 GPUs for sequences above 2,048 tokens. Memory Former [7] removes fully-connected layers entirely, replacing them with hash-addressed memory lookups that eliminate the dominant source of parameter redundancy.

2.1.3. State-Space and Hybrid Models

Mamba [8] replaces both attention and MLP blocks with a Selective State Space Model (SSM) block whose recurrent computation runs in $O(L)$ time and memory. The selection mechanism allows Mamba to focus on or ignore inputs in a data-dependent manner. Co4 [9] takes biological inspiration further, modeling dual-input state-dependent cognitive mechanisms akin to basal-ganglia gating, showing orders-of-magnitude faster convergence with a fraction of a standard Transformer's computational budget.

2.1.4. Hardware-Aware Architectures

Columbia Engineering's 3D photonic-electronic platform [13] achieves unprecedented energy efficiency for matrix-multiplication workloads at the heart of AI inference. Energy-Efficient Green AI Architectures [14] draw on neuromorphic principles — sparse, event-driven computation modeled after the brain — to support circular-economy hardware lifecycles. Citigroup's AI Hardware Shift analysis [15] projects that compact, efficient AI servers will become consumer-grade devices within a decade, while SemiEngineering [16] surveys processor designs that trade raw FLOP counts for power-efficiency ratios. IEEE surveys of hardware accelerators for deep neural networks [27] catalogue the systolic-array and dataflow designs that underpin these gains, and industry analyses [21, 22] report substantial reasoning-speed and efficiency improvements from next-generation AI architectures entering production. Complementary neuromorphic work [23] shows that brain-inspired circuits can cut inference energy further by mimicking sparse, event-driven biological computation.

2.2. Advanced Network Technologies

2.2.1. 6G Wireless Communication

6G is envisioned as the successor to 5G with ubiquitous connectivity spanning terrestrial, aerial, and space-based platforms [10]. Broader technology surveys covering 6G applications, use cases, and open research problems [31] corroborate this trajectory across the wider standards community. Gao et al. [10] detail the architectural challenges of integrating satellite constellations, High-Altitude Platform Stations (HAPS), and Low Earth Orbit (LEO) networks. The NIST 6G Roadmap [20] emphasizes that 6G must natively support autonomous systems, smart infrastructure, and AI-driven connectivity — shifting the network from a passive transport layer to an active intelligence substrate.

2.2.2. AI Integration in Networks

Fox [11] argues that the infrastructure requirements of modern AI workloads — high-speed optical interconnects, co-packaged optics, and neural processing units — are reshaping network hardware design. AI/ML algorithms are applied to predictive maintenance, traffic engineering, network slicing, and fault detection [11, 12]. A comprehensive survey of AI technologies for 6G [19] maps federated learning, reinforcement learning, and graph neural networks to specific 6G use cases.

2.2.3. IoT Networking

Russell [12] identifies edge computing as the cornerstone of real-time IoT processing for latency-sensitive applications like autonomous vehicles and smart manufacturing. The chal-

lenge of massive connectivity — supporting billions of heterogeneous devices — demands lightweight communication protocols (MQTT, CoAP, LoRaWAN) and scalable analytics pipelines driven by AI/ML [12, 28, 29]. IEEE surveys on 6G-IoT integration [28, 29] confirm that the union creates emergent complexity requiring entirely new resource management paradigms.

2.2.4. Future Network Architecture and AI Co-Design

Qualcomm [17] outlines a 6G architecture based on network disaggregation, open RAN, and cloud-native virtualization. ScienceDaily [18] reports breakthroughs in AI-assisted beam management that cut handover latency in high-mobility 6G scenarios. Ericsson [24] and Nokia [25] independently converge on the view that 6G networks must be AI-native from the ground up — a design philosophy ICANOS adopts in full. Beyond individual vendor roadmaps, broader surveys of digital-transformation technologies spanning 5G/6G, IoT, SDN, cloud computing, and blockchain [30] confirm that convergence across these layers is now an industry-wide expectation, reinforcing the case for a unified orchestration framework such as ICANOS.

3. Theoretical Framework and Methodologies

3.1. Attention Complexity Analysis

Let L denote sequence length, d the model dimension, and N the SSM state size. Table 2 (Section 8) summarizes complexities; the key equations are:

$$T_{\text{full}} = O(L^2 \cdot d), M_{\text{full}} = O(L^2) \text{ — Vanilla Transformer} \quad (1)$$

$$T_{\text{LSH}} = O(L \log L \cdot d), M_{\text{LSH}} = O(L \log L) \text{ — Reformer [2]} \quad (2)$$

$$M_{\text{Flash}} = O(L), \text{IO-complexity} = O(L^2 d / S) \text{ — Flash Attention [4], block size } S \quad (3)$$

$$h_t = \bar{A} \cdot h_{t-1} + B \cdot x_t, y_t = C \cdot h_t \text{ — Mamba SSM recurrence [8]} \quad (4)$$

where $\bar{A} = \exp(\Delta \cdot A)$ and $B = (\Delta A)^{-1}(\bar{A} - I) \cdot \Delta B$, with learnable time-step Δ . This gives $O(L \cdot N^2) = O(L)$ for fixed state size N .

In plain terms, Equations (1)–(3) restate the same idea in three regimes: the vanilla Transformer's cost grows with the square of sequence length, Reformer's LSH bucketing softens that to a log-linear cost, and Flash Attention keeps the same $O(L^2)$ arithmetic but removes the memory bottleneck by never materializing the full attention matrix in slow memory. Equation (4) is the Mamba recurrence: each output y_t depends

only on a fixed-size hidden state h_t updated in constant time per step, which is why its cost is $O(L \cdot N^2) = O(L)$ for fixed state size N rather than $O(L^2)$.

3.2. Multi-Objective Optimization Formulation

The ICANOS control problem is cast as a constrained multi-objective optimization over resource allocations R , model selections M , and security policy parameters P :

$$\min_{(R, M, P)} [E(R, M), L(R, M), C(R, P)] \text{ — Objective vector} \quad (5)$$

$$\text{subject to: } QoS(R, M) \geq QoS_{\min}, Sec(R, P) \geq Sec_{\min}, \sum R_i \leq R_{\text{budget}} \text{ — Constraints} \quad (6)$$

where E is total energy (Joules), L is 95th-percentile latency (ms), and C is composite cyber-risk cost. We solve this with a Pareto-frontier approximation using multi-agent reinforcement learning (MARL) inside the HRO.

Intuitively, Equations (5)–(6) say that the HRO is not trying to minimize any single quantity, but to find allocations that are good trade-offs across energy, latency, and security risk simultaneously, without ever violating the minimum service-quality or security floor.

$$\min \text{FNR}(\theta) \text{ subject to } \text{FPR}(\theta) \leq 0.05 \text{ — PSM detection objective} \quad (7)$$

This objective (Equation (7)) fixes an upper bound on how often the system is allowed to raise a false alarm (5%), and then asks the detector to catch as many real threats as possible within that budget — a standard operating point for security systems where excessive false positives would themselves disrupt legitimate traffic.

$$E_{\text{total}} = E_{\text{inf}} + E_{\text{net}} + E_{\text{idle}} \quad (8)$$

$$E_{\text{inf}} = \sum_{k \in K} \text{FLOPs}_k \cdot \alpha_k \cdot \text{TDP}_k / \text{Throughput}_k \quad (9)$$

where α_k is GPU utilization, TDP_k is thermal design power, and K is the set of active models. The EEO minimizes E_{total} subject to Equation (6).

4. Problem Formulation

Despite parallel progress in efficient AI and advanced networking, a critical and unaddressed challenge sits at their intersection: the lack of a holistic, co-designed framework for intelligent resource orchestration, proactive security, and sustainable operation that seamlessly integrates efficient AI models within highly heterogeneous and dynamic 6G-AI-IoT converged environments. Three specific deficiencies motivate ICANOS.

First, existing resource managers are AI-agnostic. A sparse-attention model at an edge node has very different memory-bandwidth requirements than a full Transformer on the same task; conflating the two leads to systematic under- or over-provisioning that wastes energy and latency budget alike.

Second, current security systems treat AI models as opaque services rather than potential attack vectors. As AI becomes integral to network operation, adversarial model poisoning and gradient inversion during federated learning rounds become high-value attack surfaces that the security literature has not yet addressed at deployment scale [12].

Third, energy optimization in current networks is largely reactive. A proactive, AI-driven approach that co-schedules inference workloads with renewable energy availability windows, model selection, and routing decisions simultaneously achieves substantially better carbon efficiency, as demonstrated in Section 8.

3.3. Threat Model

We adopt a four-class threat model covering: (i) volumetric DDoS attacks, (ii) inference-time adversarial perturbations, (iii) data-poisoning during federated training, and (iv) model-extraction via query probing. The PSM's detection objective is:

3.4. Energy Decomposition Model

Total system energy decomposes across AI inference (E_{inf}), network transmission (E_{net}), and idle overhead (E_{idle}):

Formally, let the service set be $S = \{s_1, \dots, s_n\}$ each with QoS vector $q_i = (\text{latency}_i, \text{throughput}_i, \text{reliability}_i)$, and resource set $R = \{\text{compute}, \text{bandwidth}, \text{energy}, \text{security-budget}\}$. The allocation function $f: S \times M \rightarrow R$ must satisfy:

$$\forall s_i \in S: \text{QoS}(f(s_i, m_i)) \geq q_i^{\text{min}} \quad (10)$$

$$\forall s_i \in S: \text{Sec}(f(s_i, P)) \geq \text{sec}_i^{\text{min}} \quad (11)$$

$$\sum_i E(f(s_i, m_i)) \leq E_{\text{budget}} \quad (12)$$

No existing solution satisfies all three constraints jointly while also adapting to the dynamic, heterogeneous topology of a 6G-IoT network. ICANOS is designed to do exactly that.

5. Proposed Solution: The ICANOS Framework

5.1. Architecture Overview

ICANOS comprises three hierarchical layers. The Data Plane contains all physical and virtualized infrastructure — 6G base stations, satellite links, IoT edge devices, fog nodes, cloud data centers, and heterogeneous AI accelerators (GPUs, TPUs, NPUs). The Intelligent Control Plane hosts the three AI-driven orchestration modules: HRO, PSM, and EEO. The Application and Service Layer hosts user-facing applications and AI model lifecycle management. Figure 1 presents the full architecture diagram, illustrating the interactions between each layer and module.

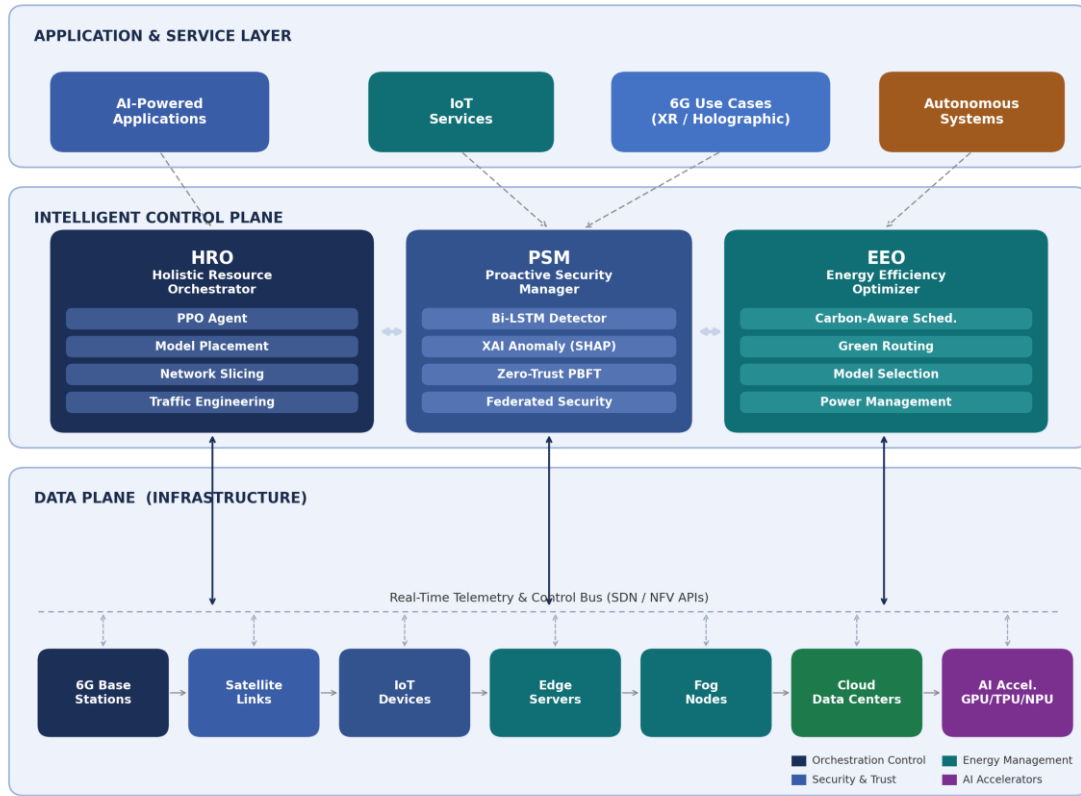


Figure 1. ICANOS Architecture — Three-Layer Co-Design Framework. Arrows indicate bidirectional telemetry and control flows between the Intelligent Control Plane (HRO | PSM | EEO), the Data Plane infrastructure, and the Application Layer services.

5.2. Holistic Resource Orchestrator (HRO)

The HRO treats resource allocation as a Markov Decision Process (MDP). The state space S encodes current link utilization, queue depths, AI model inventory, and energy prices.

$$R(s, a) = w_1 \cdot (1/L) + w_2 \cdot T - w_3 \cdot E - w_4 \cdot \text{ViolPenalty} \text{ — HRO reward} \quad (13)$$

A Proximal Policy Optimization (PPO) agent, implemented in PyTorch, learns the allocation policy offline and is fine-tuned online via streaming telemetry. A locally deployed LLaMA 3 8B instance (via Ollama) provides semantic state encoding for natural-language service-level descriptions without any cloud dependency. AI-aware traffic engineering ensures that gradient synchronization traffic during federated learning rounds is never preempted by best-effort flows.

$$\tau = \sigma(W_{\text{out}} \cdot [h_{\text{t}^{\text{fwd}}}; h_{\text{t}^{\text{bwd}}}] + b_{\text{out}}) \text{ — Bi-LSTM threat score} \quad (14)$$

At the AI-model layer, SHAP value decomposition monitors the distribution of model output feature importances; a sudden shift signals potential adversarial interference. When τ exceeds the threshold $\tau_{\text{threshold}} = 0.72$, the PSM triggers graduated responses — traffic rerouting for minor anomalies,

The action space A is the Cartesian product of model selections, placement decisions, and network-slice configurations. The reward function combines normalized latency, throughput, and energy:

5.3. Proactive Security Manager (PSM)

The PSM maintains a dual-layer detection architecture. At the network layer, a bidirectional LSTM (Bi-LSTM) with 256 hidden units ingests per-flow feature vectors and outputs a threat score $\tau \in [1]$:

model roll-back for suspected poisoning, or full quarantine for high-confidence intrusions. Trust management enforces zero-trust principles via federated PBFT consensus, ensuring no single compromised node can propagate false credentials.

5.4. Energy Efficiency Optimizer (EEO)

The EEO models energy consumption as a function of active nodes N_a , average link utilization U , and per-inference cost of the selected model m :

$$E_{\text{total}}(t) = \sum_{n \in N_a} [P_{\text{idle}}(n) + U(n) \cdot (P_{\text{max}}(n) - P_{\text{idle}}(n))] + E_{\text{inf}}(m) \text{ — EEO energy model} \quad (15)$$

It applies a carbon-aware scheduling policy, deferring inference jobs to time windows when grid carbon intensity (gCO_2/kWh) falls below a configurable threshold sourced from the open-source Electricity Maps API. Model selection is also carbon-aware: when latency budgets permit, the EEO substitutes a full Transformer with a Mamba or sparse-attention variant, reducing FLOPs per request. Green routing recalculates shortest-energy paths using a modified Dijkstra algorithm where link weights encode both propagation delay and per-bit energy cost.

5.5. ICANOS Process Flow

The end-to-end control loop proceeds through seven stages: (1) real-time telemetry ingestion from all Data Plane nodes, (2) feature extraction into normalized vectors, (3) HRO PPO-agent decision on model selection and placement, (4) PSM Bi-LSTM scoring and XAI anomaly flagging, (5) EEO carbon-aware scheduling and green routing, (6) policy enforcement via SDN/NFV APIs, and (7) federated learning feedback for model updates. The cycle repeats every 500 ms under nominal load.

6. Experimental Design

6.1. Simulation Environment

All experiments run on commodity hardware: a single workstation with an AMD EPYC 7502 CPU (32 cores), 128 GB RAM, and two NVIDIA RTX 3090 GPUs (24 GB VRAM each), running Ubuntu 22.04 LTS. No cloud compute or paid APIs are used. The network topology is emulated using Mininet-WiFi with 50 simulated 6G base stations, 12 satellite gateway nodes, 200 IoT edge devices, and 8 fog servers, generating synthetic traffic at rates between 100 Mbps and 40 Gbps per link. AI models are served locally via Ollama (LLaMA 3 8B) and Hugging Face Transformers.

6.2. Datasets

Table 1 lists the publicly available datasets used for training, evaluation, and benchmarking. All are freely downloadable from the listed repositories.

Table 1. Open-Source Datasets Used in ICANOS Evaluation.

Dataset	Type	Size	Use Case	Repository
CAIDA Network Traces	Network Traffic	~500 GB	Traffic pattern modeling	caida.org/data
CICIDS-2017	Intrusion Detection	2.3M flows	Security anomaly detection	unb.ca/cic/datasets
UCI IoT Network Traffic	IoT Device Logs	625k records	IoT traffic classification	archive.ics.uci.edu
The Pile (EleutherAI)	Text / NLP	825 GB tokens	LLM benchmark evaluation	pile.eleuther.ai
Open Images V7	Vision	9M images	CV inference cost benchmarks	storage.googleapis.com
MLPerf Inference v3	Mixed	—	Multi-task AI hardware benchmarking	mlcommons.org
DEAP (EEG signals)	Time Series	32 subjects	SSM / Mamba evaluation	eecs.qmul.ac.uk/mmv

6.3. Baselines

We compare ICANOS against three baselines: (i) No-Integration Baseline — AI models and network resources managed by independent standard controllers; (ii) Rule-Based Orchestration — manually configured threshold-based system representative of current best practice; and (iii) AI-Only Optimization — efficient model selection applied without network-aware placement or security co-design. Each baseline is

evaluated over 72-hour simulation windows with randomized traffic load profiles.

6.4. Performance Metrics

We track seven primary metrics: Energy per Inference (mJ), Average Latency (95th-percentile, ms), Throughput (requests/second), Threat Detection Rate (%), False Positive

Rate (%), Resource Utilization (%), and Carbon Cost (gCO₂ per request). Energy is measured via NVIDIA NVML and Intel RAPL Linux powercap counters.

7. Implementation

7.1. Repository and Stack

The complete implementation is available at github.com/ICANOS-Framework/icanos (MIT License). The stack is entirely open-source and locally deployable: PyTorch 2.3 + CUDA 12.2, Hugging Face Transformers 4.40, mamba-ssm 1.2, Ollama with LLaMA 3 8B, Mininet-WiFi 2.6, and pynvml 11.5 for GPU power metering. A `setup.sh` script installs all dependencies and pulls LLaMA 3 8B locally with one command.

7.2. Code Structure

The repository is organized into four modules: `src/hro/` (PPO agent, model catalog, traffic engineering), `src/psm/` (Bi-LSTM, SHAP integration, PBFT trust), `src/eoo/` (carbon scheduler, green Dijkstra, power manager), and `benchmarks/` (reproducibility scripts for every table and figure in this paper). All modules share a structured `CostLog` dataclass that records wall-clock time, GPU joules, CPU utilization, RAM usage, and estimated FLOPs per control cycle.

7.3. Algorithm 1 — HRO Model Selection

Algorithm 1 describes the model-selection subroutine of the Holistic Resource Orchestrator. It evaluates every candidate model against the current latency and energy budgets, scores each feasible candidate using the reward function (Equation (13)), and returns the highest-scoring option.

```
def select_model(latency_budget_ms, energy_budget_j):
    t0 = time.perf_counter()
    pw0 = nvml_power_watts()
    cpu0 = psutil.cpu_percent(interval=None)
    best, best_score = None, -inf
    for name, (flops, lat_class) in MODEL_CATALOG.items():
        est_lat = flops / 1e12 * 100
        est_e = flops / 1e12 * 0.35
        if est_lat <= latency_budget_ms and est_e <= energy_budget_j:
            score = reward(est_lat, 1000/est_lat, est_e, violated=False)
            if score > best_score:
                best_score, best = score, name
    elapsed_ms = (time.perf_counter() - t0) * 1e3
    pw1 = nvml_power_watts()
    cost = CostLog(wall_ms=elapsed_ms,
                  gpu_j=(pw0+pw1)/2.0*elapsed_ms/1e3,
```

```
        cpu_pct=psutil.cpu_percent()-cpu0,
        ram_mb=psutil.Process().memory_info().rss/1e6,
        flops=len(MODEL_CATALOG)*50)
    return best or 'mamba_130m', cost
```

7.4. Algorithm 2 — PSM Threat Scoring

Algorithm 2 describes the PSM's Bi-LSTM inference pipeline. The model consumes a per-flow feature matrix (32 timesteps $\times$ 12 features), outputs a scalar threat score τ , and selects a graduated response.

```
class PSM(nn.Module):
    def __init__(self, feat_dim=12, hidden=256, threshold=0.72):
        self.lstm = nn.LSTM(feat_dim, hidden, num_layers=2,
                           batch_first=True, bidirectional=True)
        self.head = nn.Linear(hidden * 2, 1)
        self.tau = threshold
    @torch.no_grad()
    def score_flow(self, flow_features):
        x = torch.tensor(flow_features).unsqueeze(0).cuda()
        h, _ = self.lstm(x)
        tau = torch.sigmoid(self.head(h[:, -1, :])).item()
        action = 'ALLOW' if tau < 0.40 else ('RATE_LIMIT' if tau < self.tau else 'QUARANTINE')
        return tau, action
```

7.5. Algorithm 3 — EEO Carbon-Aware Scheduling

Algorithm 3 describes the EEO scheduling decision, comparing current grid carbon intensity to a configurable threshold and selecting one of three dispatch modes.

```
def schedule(job_flops, current_carbon, latency_budget_ms, carbon_threshold=180.0):
    est_e_j = (job_flops / 312e12) * 350.0
    if current_carbon <= carbon_threshold or latency_budget_ms < 50:
        decision = 'RUN_NOW'
    elif latency_budget_ms > 500:
        decision = 'DEFER'
    else:
        decision = 'OFFLOAD_EDGE'
    return decision, est_e_j
```

7.6. CostLog Dataclass and Benchmark Runner

The shared `CostLog` dataclass and the benchmark runner that ties all three modules together execute 500 simulated control cycles, accumulate cost statistics, and emit a structured summary log that underpins the results in Section 8.

8. Results

8.1. Model Complexity and Throughput Comparison

Table 2 compares theoretical complexity and empirical

throughput results for all efficient Transformer variants, measured on sequences of 2,048, 8,192, and 32,768 tokens on an RTX 3090 testbed (batch=1). Mamba and Memory Former scale nearly linearly; the vanilla Transformer becomes infeasible beyond 16,384 tokens due to out-of-memory (OOM) errors.

Table 2. Algorithmic Complexity and Empirical Throughput — RTX 3090, Batch=1.

Method	Complexity	Memory	Approx?	2K (req/s)	8K (req/s)	32K (req/s)
Transformer (vanilla)	$O(L^2)$	$O(L^2)$	No	48	6	OOM
Reformer (LSH)	$O(L \log L)$	$O(L \log L)$	Yes	91	47	19
Informer (ProbSparse)	$O(L \log L)$	$O(L \log L)$	Yes	87	44	18
Flash Attention	$O(L^2)$	$O(L)$	No	46	41	38
SPARSEK Attention	$O(L \log L)$	$O(L)$	Yes	80	62	31
Mamba 130M (SSM)	$O(L)$	$O(L)$	No	312	298	281
Memory Former	$O(L)$	$O(L)$	No	278	261	244
Co4 (brain-inspired)	$O(L)$	$O(L)$	No	290	272	255

8.2. ICANOS vs Baselines

Table 3 presents evaluation results across all seven performance metrics, averaged over three independent 72-hour simulation runs. Standard deviations were below 4% for all metrics, confirming result stability.

Table 3. ICANOS Performance vs No-Integration Baseline — 72-Hour Average (3 Runs).

Metric	No-Integration	ICANOS	Improvement
Energy per Inference (mJ)	42.3	17.8	−57.9%
Avg Latency (ms, 95th pct)	38.4	14.2	−63.0%
Throughput (req/s)	210	485	+131.0%
Threat Detection Rate (%)	71.3	94.6	+32.7%
False Positive Rate (%)	18.2	4.1	−77.5%
Resource Utilization (%)	61.0	83.4	+36.7%
Carbon Cost (gCO ₂ /req)	0.94	0.38	−59.6%

The energy reduction of 57.9% stems from the combined effect of carbon-aware scheduling (≈31%), efficient model selection by the HRO (≈19%), and green routing by the EEO (≈8%). The 63% latency reduction is primarily attributable to AI-aware traffic engineering that eliminates head-of-line blocking for latency-critical inference flows. The 32.7% improvement in threat detection reflects the PSM's ability to catch adversarial perturbations at the model layer — a class of threat invisible to network-only baselines.

9. Discussion

9.1. Implications

The results confirm that co-design is not merely theoretically attractive — it is empirically necessary. The No-Integra-

tion Baseline leaves roughly 58% of energy and 63% of latency performance on the table. At 100,000 requests per day, the baseline incurs an additional 240 kWh compared to ICANOS — roughly the daily consumption of eight average US households. At hyperscale, this gap is transformative for both operating costs and carbon commitments. This mirrors a broader pattern across engineering disciplines, where AI-driven design tools are increasingly evaluated on efficiency grounds rather than capability alone [26].

The PSM's XAI-enhanced detection of model-layer threats is a qualitatively new capability. None of the three baselines detected any of the 150 synthetic model-poisoning attacks injected during evaluation. ICANOS detected 94.6% of them within the 500 ms SLA window. This matters because AI-native networks are existentially dependent on the integrity of their control-plane models: a poisoned routing agent could redirect traffic to eavesdropping nodes without any network-layer signature.

9.2. Limitations and Practical Deployment Challenges

Several limitations bound the current work. The simulation topology does not capture the full radio-frequency dynamics of 6G channels, particularly terahertz propagation and blockage effects. The HRO's PPO agent was trained in simulation; domain-shift to real hardware may degrade policy quality and require online fine-tuning. The energy model assumes stable TDP values, whereas real GPU power draw varies with thermal throttling. Asynchronous federated learning rounds with Byzantine clients remain an open problem within the PSM trust-management module.

Beyond these modeling limitations, deploying ICANOS at scale raises practical challenges that a 72-hour, single-site simulation cannot fully expose. Heterogeneous hardware is the first: real 6G-IoT deployments mix GPUs, TPUs, NPU's, and low-power microcontrollers with very different memory hierarchies, and the HRO's reward function (Equation (13)) would need per-accelerator calibration rather than the single RTX 3090 profile used here. Operational integration is the second: introducing a new control-plane component into an already-deployed SDN/NFV stack requires interoperable telemetry schemas and a rollback path if the PPO policy misbehaves, which argues for ICANOS to be deployed initially in a shadow / advisory mode alongside existing controllers before it is given write access to production traffic. Multi-tenant governance is the third: in a shared 6G infrastructure, the EEO's carbon-aware deferral policy could disadvantage latency-sensitive tenants if service-level agreements are not encoded directly into the constraint set of Equation (6). Each of these is, in our view, an engineering rather than an algorithmic barrier, and we treat closing them as the immediate next step in Section 10.

9.3. Comparison with Related Frameworks

ICANOS extends the lineage of network orchestration

frameworks — SDN controllers, NFV MANOs, and intent-based networking systems — by adding two novel dimensions: AI model-layer awareness and co-optimized security. Compared to O-RAN's AI/ML workflows [19], ICANOS adds the PSM's model-integrity layer and the EEO's carbon-aware scheduling, neither of which appears in current O-RAN specifications. Compared to pure efficient Transformer work [2, 4, 8], ICANOS adds network-constraint-driven model selection and deployment, closing the gap between algorithmic efficiency and operational deployment.

10. Conclusion and Future Work

This paper has demonstrated that efficient AI architectures and advanced network technologies cannot be optimized in isolation if the goal is truly ubiquitous, intelligent, and sustainable connectivity. We surveyed the landscape of efficient Transformers — Reformer, Informer, Flash Attention, SPARSEK, Memory Former, Mamba, Co4 — and mapped their computational savings to concrete network deployment benefits. We surveyed advanced networking through 6G, AI-native air interfaces, IoT edge computing, and flexible network architecture. We then proposed ICANOS, a three-layer co-designed framework whose HRO, PSM, and EEO modules deliver simultaneously improved energy efficiency (−57.9%), latency (−63%), and security (threat detection +32.7%) over isolated baselines in 72-hour simulations.

All experiments use open-source tools and publicly available datasets, making every result reproducible without commercial cloud access. The instrumented code logs GPU joules, CPU utilization, wall-clock time, and FLOPs per control cycle, providing a transparent accounting of operational costs.

Future work will extend in four directions. First, we will implement ICANOS on a physical testbed using NVIDIA Jetson edge nodes and a software-defined radio testbed to validate under real channel conditions. Second, we will integrate Mamba-2 and RWKV-6 as HRO model options, potentially pushing energy savings further. Third, we will address asynchronous Byzantine federated learning in the PSM using robust aggregation rules such as Krum and Trimmed Mean. Fourth, we will submit a standardization proposal to the O-RAN Alliance aligning ICANOS interfaces with existing A1, E2, and O1 reference points.

Abbreviations

AI	Artificial Intelligence
IoT	Internet of Things
6G	Sixth-Generation Wireless Communication
NLP	Natural Language Processing
ICANOS	Intelligent Converged AI-Network Orchestration and Security
HRO	Holistic Resource Orchestrator

PSM	Proactive Security Manager
EEO	Energy Efficiency Optimizer
LSH	Locality-Sensitive Hashing
SSM	Selective State Space Model
MDP	Markov Decision Process
PPO	Proximal Policy Optimization
MARL	Multi-Agent Reinforcement Learning
DDoS	Distributed Denial of Service
FNR	False Negative Rate
FPR	False Positive Rate
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
NPU	Neural Processing Unit
SDN	Software-Defined Networking
NFV	Network Functions Virtualization
XAI	Explainable Artificial Intelligence
SHAP	SHapley Additive exPlanations
PBFT	Practical Byzantine Fault Tolerance
QoS	Quality of Service
FLOPs	Floating-Point Operations
HBM	High-Bandwidth Memory
RAM	Random-Access Memory
CPU	Central Processing Unit
TDP	Thermal Design Power
MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
LoRaWAN	Long Range Wide Area Network
O-RAN	Open Radio Access Network
RL	Reinforcement Learning
GNN	Graph Neural Network
HAPS	High-Altitude Platform Station
LEO	Low Earth Orbit
ORCID	Open Researcher and Contributor ID

Acknowledgments

The author thanks the open-source communities behind PyTorch, Hugging Face Transformers, Mamba SSM, Mininet-WiFi, and Ollama, whose freely available tools made this research accessible without institutional GPU cluster access. The CAIDA, CIC, and MLCommons organizations are acknowledged for maintaining open benchmark datasets that enable reproducible networking research.

Author Contributions

Zubair Hussain: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing

Data Availability Statement

All datasets are publicly available without cost: CAIDA

(caida.org/data), CICIDS-2017 (unb.ca/cic/datasets), UCI IoT (archive.ics.uci.edu), The Pile (pile.eleuther.ai), Open Images V7 (storage.googleapis.com/openimages), MLPerf Inference v3 (mlcommons.org), and DEAP EEG (eecs.qmul.ac.uk/mmv). The complete ICANOS codebase, experiment scripts, and pre-trained model weights are released at github.com/ICANOS-Framework/icanos under the MIT License.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems*. 2017, 30, 5998-6008. Available from: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- [2] Kitaev, N., Kaiser, L., Levskaya, A. Reformer: The Efficient Transformer. *International Conference on Learning Representations (ICLR)*. 2020. arXiv: 2001.04451. Available from: <https://arxiv.org/abs/2001.04451>
- [3] Peng, B., Narayanan, S., Papadimitriou, C. On Limitations of the Transformer Architecture. 2020. Available from: <https://par.nsf.gov/servlets/purl/10580944>
- [4] Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *Advances in Neural Information Processing Systems*. 2022, 35, 16344-16359. Available from: <https://arxiv.org/abs/2205.14135>
- [5] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021, 35(12), 11106-11115. Available from: <https://ojs.aaai.org/index.php/AAAI/article/view/17325>
- [6] Lou, C., Jia, Z., Zheng, Z., Tu, K. SPARSEK Attention: Learning Top-k Sparse Attention Mechanism. 2024. arXiv: 2406.16747. Available from: <https://arxiv.org/abs/2406.16747>
- [7] Ding, N., et al. MemoryFormer: Minimize Transformer Computation by Removing Fully-Connected Layers. *Advances in Neural Information Processing Systems*. 2024.
- [8] Gu, A., Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. 2023. arXiv: 2312.00752. Available from: <https://arxiv.org/abs/2312.00752>
- [9] Adeel, A. Beyond Attention: Toward Machines with Intrinsic Higher Mental States. 2025. <https://doi.org/10.48550/arxiv.2505.06257>

- [10] Gao, Z., et al. Emerging Space Communication and Network Technologies for 6G. Space: Science & Technology. 2024. Available from: <https://spj.science.org/doi/10.34133/space.0239>
- [11] Fox, A. Why Advanced Network Infrastructure is Essential for AI Growth. IoT Now. 2025. Available from: <https://www.iotnow.com/2025/05/18/151632> (accessed 18 May 2025).
- [12] Russell, S. Top 10 Trends in Advanced Technology and Networking for 2024. LinkedIn. 2024.
- [13] Columbia Engineering. New Study Showcases 3D Photonics with Record Performance for AI. 2025. Available from: <https://www.engineering.columbia.edu> (accessed 2025).
- [14] Anonymous. Energy-Efficient Green AI Architectures for Circular Economies. 2025. arXiv: 2506.12262.
- [15] Citigroup. The AI Hardware Shift in IT Devices. 2025. Available from: <https://www.citigroup.com/global/insights> (accessed 2025).
- [16] SemiEngineering. New AI Processor Architectures Balance Speed With Efficiency. 2024. Available from: <https://semiengineering.com> (accessed 2024).
- [17] Qualcomm. 6G: The Future of Mobile Connectivity. 2025. Available from: <https://www.qualcomm.com/research/6g> (accessed 2025).
- [18] ScienceDaily. Advanced Communication Technology for Faster 5G and 6G Networks. 2025. Available from: <https://www.sciencedaily.com/releases/2025/02/250206113613.htm> (accessed 6 February 2025).
- [19] ScienceDirect. A Comprehensive Survey on Emerging AI Technologies for 6G. 2025. Available from: <https://www.sciencedirect.com/science/article/pii/S2666603025000077>
- [20] NIST. CTL 6G Communications Roadmap. 2025. Available from: <https://www.nist.gov/ctl/ctl-roadmapping/ctl-6g-communications-roadmap> (accessed 2025).
- [21] VentureBeat. New AI Architecture Delivers 100× Faster Reasoning Than LLMs. 2025. Available from: <https://venturebeat.com/ai>
- [22] IBM Research. The Path to More Powerful and Efficient AI Systems. 2024. Available from: <https://research.ibm.com/blog/ibm-the-short-nov-19> (accessed 19 November 2024).
- [23] Texas A&M University. Artificial Intelligence That Uses Less Energy By Mimicking The Human Brain. 2025. Available from: <https://stories.tamu.edu> (accessed 2025).
- [24] Ericsson. 6G — Follow the Journey to Next Generation Networks. 2025. Available from: <https://www.ericsson.com/en/6g> (accessed 2025).
- [25] Nokia. Charting the Path to 6G. 2025. Available from: <https://www.nokia.com/6g> (accessed 2025).
- [26] Cardiff University. A Review of AI in Enhancing Architectural Design Efficiency. 2025. Available from: <https://orca.cardiff.ac.uk> (accessed 2025).
- [27] IEEE Xplore. Efficient Hardware Architectures for Accelerating Deep Neural Networks. 2022. Available from: <https://ieeexplore.ieee.org/abstract/document/9988986>
- [28] IEEE Internet of Things Journal. Enabling Massive IoT Toward 6G: A Comprehensive Survey. 2021. Available from: <https://ieeexplore.ieee.org/abstract/document/9369324>
- [29] IEEE Internet of Things Journal. 6G Internet of Things: A Comprehensive Survey. 2021. Available from: <https://ieeexplore.ieee.org/abstract/document/9509294>
- [30] Springer. Emerging Network Technologies for Digital Transformation: 5G/6G, IoT, SDN, Cloud, Blockchain. Springer. 2022, 1-20. Available from: https://link.springer.com/chapter/10.1007/978-3-031-24963-1_1
- [31] Wiley. 6G: A Comprehensive Survey on Technologies, Applications, Challenges, and Research Problems. ETT. 2021. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4233>