BEAT: An Integrated Blockchain-Edge AI Trust Framework for IoT Security - Design, Prototype, and Performance Evaluation

Pavansai Ramarao Maddali, Vijay Kumar Damera, Ratna Kumar Prathipati
Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Aziznagar, Hyderabad - 500075, Telangana, India

***Abstract*—Trust management in heterogeneous Internet of Things (IoT) deployments remains unresolved because most frameworks treat blockchain reputation persistence and edge-AI anomaly detection as separate subsystems, so the graded evidence a detection model produces cannot be indexed or propagated by a ledger that was built without reference to it. This study presents BEAT (Blockchain–Edge AI Trust), which closes that gap through co-design of three coupled layers: a Graph Attention Network–Long Short-Term Memory (GAT-LSTM) edge module that converts device interaction graphs into normalised trust evidence vectors; a Hyperledger Fabric 2.5 ledger whose TrustCC chaincode enforces monotone trust updates over a formally defined trust lattice; and a Proximal Policy Optimisation (PPO) admission controller whose advantage over contextual-bandit and PID baselines is established both theoretically and experimentally. A companion gossip protocol, GARP, propagates reputation deltas across edge orchestrators with a proven logarithmic convergence guarantee. Prototyped on a 12-node Raspberry Pi 5 cluster attached to a three-peer Fabric channel and evaluated on UNSW-NB15, N-BaIoT, TON_IoT, and CICIoT2023, BEAT reaches a cross-dataset mean F1 of 0.961, an on-chain throughput of 1,847 transactions per second, and a median trust-decision latency of 38 ms. BEAT outperforms a GNN-Transformer baseline (F1 0.944) and a CNN-BiLSTM-Transformer baseline (F1 0.939) on UNSW-NB15, and ablation results show that removing any single BEAT layer costs at least four F1 points cross-dataset, evidence that the system-level integration, not the detection model alone, drives the gain. Formal Byzantine resilience bounds accompany the prototype evaluation, alongside experimental Sybil-resistance and evidence-poisoning results.**

***Keywords*—Blockchain trust management; edge AI security; GAT-LSTM anomaly detection; Hyperledger Fabric; gossip-accelerated reputation; IoT admission control; Byzantine fault tolerance**

I. Introduction

The proliferation of Internet of Things (IoT) devices—conservatively estimated at 18.8 billion by the end of 2024 and projected to exceed 38 billion by 2030 [1]—has transformed a theoretical security concern into an operational reality. A compromised sensor in an industrial process can silently inject falsified readings for hours; a Mirai-infected thermostat becomes a DDoS relay within seconds of botnet contact [2]. Classical cryptographic authentication addresses identity verification but cannot answer the question that operations teams most need answered after a device has been admitted to a network: "Should I still trust what this device is reporting?"

Trust management attempts to answer that question by accumulating behavioural evidence and computing per-device trust scores that reflect recent conduct, not merely credential possession [3]. The two most natural substrates for trust management in distributed IoT environments are blockchain—for tamper-resistant, auditable evidence storage—and edge AI—for low-latency, high-accuracy behavioural inference. Both have been explored extensively in isolation. What has not been engineered is their *co-design*: an architecture in which the output representation of the AI layer is the input representation of the ledger layer, designed from the outset to be semantically compatible.

This co-design problem is not just an integration question; it carries formal constraints. The trust evidence vector that the AI layer produces must be compact enough for on-chain storage, structured so that chaincode can enforce monotone update semantics without floating-point non-determinism across peers, and expressive enough that admission-threshold policies can be derived from it without losing information. Meeting all three constraints simultaneously requires designing the inference architecture, the smart contract logic, and the propagation protocol together—not composing them post hoc.

Existing frameworks fall short on at least one constraint. EdgeGuard-IoT [4] stores data hashes rather than trust scores, forfeiting gradation. AI-SET [5] does not persist inference outputs at all. TX-BFT [6] maintains graded reputations but uses only rule-derived trust evidence, not AI inference. HLF-FSL [7] achieves strong federated confidentiality but does not implement trust management. None of these is a complete solution; each addresses a subset of the co-design constraints.

A. Precise Novelty Statement

BEAT's novelty resides in three contributions that no prior work combines: (1) a formally defined trust lattice that is the shared semantic specification for both the GAT-LSTM inference layer and the TrustCC chaincode, ensuring the two cannot diverge; (2) GARP, a gossip protocol engineered to propagate trust deltas at the speed demanded by IoT dynamics without adding network overhead, proved formally here; and (3) a PPO-based admission controller whose advantage over contextual-bandit and PID alternatives is demonstrated experimentally across four datasets under realistic non-stationary attack conditions. These three contributions are individually modest refinements of known techniques; their value is in the particular system-level integration discipline they enforce, which has measurable consequences for security performance.

Contributions. This study makes the following specific claims:

1) Co-designed trust semantics: We define a complete trust lattice $(\mathcal{L}, \preceq)$ and prove that BEAT's trust update function is a monotone self-map on $\mathcal{L}$ (Theorem 1), establishing that the chain and the inference module are formally coupled.
2) GARP with convergence proof: We prove that GARP achieves globally consistent reputation views in $O(\log N)$ gossip rounds with failure probability $\leq N^{-2}$ under churn $\rho < 0.15$ and connectivity $p_c > 0.8$ (Theorem 2, full proof in Appendix A).
3) PPO superiority with regret analysis: We prove a sub-linear policy regret bound for the PPO controller under L-smooth, non-stationary attack-intensity dynamics (Theorem 3), and empirically verify PPO's advantage over contextual-bandit UCB and PID baselines in four attack scenarios.
4) Byzantine resilience bound: We derive an upper bound on the trust corruption achievable by f Byzantine edge orchestrators under BEAT's multi-endorser update model (Proposition 1), and validate the bound experimentally by injecting Byzantine peers.
5) Cross-dataset generalisation: BEAT is trained on UNSW-NB15 and evaluated without retraining on N-BaIoT, TON_IoT, and CICIoT2023, establishing cross-domain generalisability rather than single-benchmark performance.
6) Open prototype: All code, chaincode, trained weights, and experiment scripts will be publicly archived at GitHub (https://github.com/dameravijay/BEAT-framework) upon acceptance, with a Docker-based emulation for hardware-free reproduction.

II. Related Work

A. Blockchain-Based Trust and Reputation

Liu *et al.* [8] provide the most comprehensive taxonomy of blockchain-based IoT trust management, identifying a key limitation that motivates BEAT directly: continuous-score schemes on permissioned chains are the most expressive, yet no existing work at the time of their survey had solved the evidence-ingestion problem—how to translate behavioral observations at the edge into chain-compatible inputs without bandwidth bottlenecks. BEAT's trust evidence vector and GARP protocol together constitute a response to precisely this gap.

Tyagi *et al.* [9] survey trust management techniques across IoT, classifying approaches by entity type, evidence collection method, and update mechanism. Their analysis reveals that methods relying on direct observation alone scale poorly, while those that propagate indirect (recommendation-based) evidence are susceptible to collusion—a finding that motivates BEAT's hybrid of direct GAT-LSTM inference and gossip-propagated reputation.

Nguyen *et al.* [10] explore the broader integration landscape of edge computing and blockchain for IoT, noting architectural tensions between Fabric's ordering latency and the millisecond decision cycles required for real-time admission control. BEAT resolves this tension by decoupling inference (edge-local, sub-20 ms) from commitment (on-chain, asynchronous) and using the PPO agent to absorb the scheduling jitter.

Begum *et al.* [11] integrate blockchain with federated learning for intrusion detection, achieving 96.3% accuracy but storing only binary intrusion flags on-chain. This forfeits the graded trust information that BEAT preserves, limiting downstream reasoning about device reliability gradient.

B. GNN and Transformer Baselines for IoT Intrusion Detection

Recent SOTA work has moved well beyond CNN-LSTM architectures. Zhang *et al.* [12] propose CNN-BiLSTM-Transformer on CIC-IDS2017 and BoT-IoT, achieving 99.80% and 97.95% accuracy, respectively, using XGBoost+Chi-squared feature selection and Borderline-SMOTE. This work is our primary detection-layer upper-bound reference.

E-T-GraphSAGE [13] fuses GraphSAGE with Transformer self-attention, achieving strong performance on network traffic graphs but requiring a static graph structure incompatible with BEAT's dynamic $\mathcal{G}_t$ construction. Its lack of blockchain persistence and admission control make direct comparison partial—we include it as a *detection-only* baseline.

The GNN-Transformer hybrid of Bao *et al.* [14] applies Gray Level Co-occurrence Matrix texture fusion with GNN for IDS, achieving competitive F1 on UNSW-NB15. Again, the absence of trust persistence and adaptive thresholding confines this to the detection layer of our comparison hierarchy.

The GAT-based IoT IDS of Rashid *et al.* [15] demonstrates that graph attention—specifically—offers structural advantages for IoT topology modelling compared to SAGE-style neighbour sampling, validating BEAT's choice of GAT over GraphSAGE.

1) Why BEAT does not adopt a Transformer inference layer?: Transformer self-attention is $O(T^2 d)$ in sequence length T and hidden dimension d. On Raspberry Pi 5 with 8 GB RAM, a 24-timestep Transformer layer requires 31 ms inference latency versus 14 ms for the GAT-LSTM combination (Section VI), violating the 38 ms total-budget constraint. The F1 gap between GAT-LSTM (0.973) and the Transformer baseline (0.939) additionally demonstrates that BEAT's graph-structural encoding is *more* informative than self-attention for IoT device interaction topology, not merely more efficient.

C. Adaptive Admission Control and Reinforcement Learning

Ngo *et al.* [16] demonstrate contextual-bandit admission control in hierarchical edge computing, achieving a good accuracy-delay trade-off for single-step anomaly detection decisions. Contextual bandits are a natural comparison for BEAT's PPO controller because they require fewer samples and have $O(\sqrt{T})$ regret under stationarity. The argument for PPO rather than bandits in BEAT is that IoT admission control is a *multi-step sequential* decision problem—threshold adjustments interact across cycles because a raised threshold blocks devices whose subsequent behavior would have exonerated them—and PPO's full policy gradient handles this temporal credit assignment while bandits cannot. Section VI validates this experimentally.

Liu *et al.* [17] use deep reinforcement learning to optimise PBFT consensus parameters in blockchain-enabled IIoT, an approach complementary to BEAT's PPO admission controller. Their work does not address trust score management or adaptive thresholding, but it confirms the viability of RL for chain parameter control in IoT contexts.

D. Federated Learning and Blockchain

The ABAC-FL framework [18] uses Hyperledger Fabric smart contracts to control FL round participation via attribute-based access policies, reducing model poisoning attack surfaces. HPoT [19] introduces hierarchical proof-of-trust for IIoT federated learning with Byzantine-robust weighted averaging, achieving 94.2% accuracy on heterogeneous data—closely related in motivation to BEAT's Byzantine resilience analysis (Section III). The key structural difference: both works control *model contribution* rights, while BEAT controls device *operational admission* rights based on continuous trust scores.

HLF-FSL [7] achieves state-of-the-art accuracy parity with centralised split learning on CIFAR-10 and ImageNet-Mini using Hyperledger Fabric Private Data Collections. Its chaincode engineering provides several insights adopted in BEAT's TrustCC implementation.

E. Precise Positioning

Table I positions BEAT against ten recent systems across eight dimensions. BEAT is the only system that simultaneously satisfies all eight: on-chain continuous reputation, AI-based evidence generation, formal trust model, adaptive thresholding, SOTA AI baseline comparison, cross-dataset evaluation, formal convergence proof, and open artefacts.

III. Formal Trust Model and Theoretical Analysis

A. Trust Lattice and Formal Definitions

Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ be the device population and $\mathcal{G}_t = (\mathcal{D}, E_t, \mathbf{X}_t)$ a weighted directed interaction graph at time t, where $\mathbf{X}_t \in \mathbb{R}^{N\times p}$ is the matrix of per-device feature vectors and E_t contains directed edges weighted by communication intensity.

Definition 1 (Trust Score)**.** *The trust score of d_i at time t is $\tau_i(t) \in [0,1]$, with $\tau_i(0) = 0$ for all newly joined devices. A device is provisionally trusted if $\tau_i(t) \in [\theta_k, 1]$ for the current domain threshold θ_k, and quarantined otherwise.*

Definition 2 (Trust Evidence Vector)**.** *The trust evidence vector $\mathbf{e}_i(t) \in [0,1]^k$ ($k = 8$) is the projection of the GAT-LSTM output for device d_i at time t. Component indices are: anomaly score e_1, packet-rate deviation e_2, inter-arrival entropy e_3, payload entropy e_4, peer-reputation neighbourhood mean e_5, protocol-distribution KL divergence from baseline e_6, denial-of-service indicator e_7, and recovery flag e_8. All components are sigmoid-projected to $[0,1]$, to satisfy on-chain range constraints.*

Definition 3 (Trust Update Function)**.**

$$\tau_i(t{+}1) \;=\; f\big(\tau_i(t),\, \mathbf{e}_i(t)\big) \;=\; \alpha\,\tau_i(t) + (1{-}\alpha)\,\sigma\big(\mathbf{w}^\top \mathbf{e}_i(t){+}b\big), \tag{1}$$

where, $\alpha \in (0,1)$ is a temporal momentum factor, $\sigma(\cdot)$ is the logistic sigmoid, and $(\mathbf{w}, b) \in \mathbb{R}^k \times \mathbb{R}$ are the learned parameters of the GAT-LSTM projection head.

Definition 4 (Trust Lattice)**.** *The complete lattice $\mathcal{L} = ([0,1], \leq, \wedge, \vee, 0, 1)$ with $\tau \wedge \tau' = \min(\tau, \tau')$ and $\tau \vee \tau' = \max(\tau, \tau')$ serves as the semantic domain for all trust operations in BEAT. TrustCC chaincode stores and updates trust scores exclusively within $\mathcal{L}$.*

B. Monotonicity and Contraction

Theorem 1 (Monotone Trust Update)**.** *Let $\mathbf{e}_i(t)$ be fixed. Then $f(\cdot, \mathbf{e}_i(t))$ is a monotone non-decreasing function of $\tau_i(t)$. Moreover, f is a strict contraction on $\mathcal{L}$ with Lipschitz constant $\alpha < 1$, so f has a unique fixed point $\tau_i^* \in \mathcal{L}$ for any fixed evidence $\mathbf{e}_i$.*

Proof: Differentiating Eq. (1) with respect to $\tau_i(t)$ yields $\partial f/\partial \tau_i = \alpha \in (0,1)$, which is positive, establishing monotonicity. For any $\tau, \tau' \in [0,1]$:

$$|f(\tau, \mathbf{e}) - f(\tau', \mathbf{e})| = \alpha|\tau - \tau'|.$$

Since $\alpha < 1$, $f(\cdot, \mathbf{e})$ is a strict contraction on $([0,1], |\cdot|)$. By the Banach Fixed Point Theorem, it has a unique fixed point $\tau_i^* = \sigma(\mathbf{w}^\top \mathbf{e}_i + b)$, obtained by setting $\tau_i(t+1) = \tau_i(t)$ in Eq. (1). ∎

Corollary 1 (Convergence Rate)**.** *Starting from any $\tau_i(0) \in [0,1]$, the iterates $\{\tau_i(t)\}_{t\geq 0}$ converge to τ_i^* at an exponential rate: $|\tau_i(t) - \tau_i^*| \leq \alpha^t |\tau_i(0) - \tau_i^*|$. For $\alpha = 0.7$, the trust score is within 1% of τ_i^* after at most $t^* = \lceil \log(0.01)/\log(0.7) \rceil = 13$ update cycles.*

This convergence rate is operationally meaningful: a newly joined device reaches a stable trust score within 13 observation windows (13 minutes at the default 60-second cycle), after which its admission status reflects a genuinely accumulated behavioral record.

C. GARP Convergence

Consider M edge orchestrators, each maintaining a local trust view $\mathcal{T}_v = \{\tau_i^{(v)}\}$. GARP propagates trust deltas $\delta_i = \tau_i^{\text{new}} - \tau_i^{\text{old}}$ by piggybacking on IoT heartbeat packets already scheduled every $\Delta_h = 30$ seconds.

Theorem 2 (GARP Convergence)**.** *Under peer churn rate $\rho < 0.15$ and inter-orchestrator connectivity $p_c > 0.8$, with each informed orchestrator gossiping to $\beta = \lceil 2 \ln M / p_c \rceil$ random peers per round, GARP achieves a globally consistent trust view across all M orchestrators in $r^* = O(\log M)$ rounds with probability at least $1 - M^{-2}$. Bandwidth overhead per orchestrator per round is $O(\beta \cdot k \cdot |\{i : |\delta_i| > \varepsilon_{\min}\}|)$ bytes, where $k = 8$ is the evidence dimension and $\varepsilon_{\min} = 0.02$ is the minimum reportable delta.*

Proof: See Appendix A. The proof applies a push-gossip Chernoff argument, tracking uninformed orchestrators across rounds and bounding their expected count below 1 after $r^* = \lceil (2 \ln M + 2 \ln \ln M)/p_c \rceil$ rounds. ∎

Remark 1. At $M = 7$ orchestrators (our testbed) with $p_c = 1.0$ (full Ethernet connectivity), $r^* = 3$ rounds—confirmed by

TABLE I. COMPARATIVE POSITIONING OF BEAT AGAINST RECENT RELATED WORK. ✓ = FULLY ADDRESSED; ∼ = PARTIALLY ADDRESSED; × = NOT ADDRESSED. **BOLD**: BEAT'S ADVANTAGE OVER THE SPECIFIC SYSTEM IN THE MOST CLOSELY RELATED DIMENSION.

System	On-chain Cont. Rep.	AI-based Evidence	Formal Trust	Adaptive Threshold	SOTA Baselines	Cross-dataset Eval.	Convergence Proof	Open Artefacts
Begum [11]	Binary	FL	×	×	∼	×	×	✓
Wardana [20]	×	Lightweight	×	×	×	×	×	✓
HLF-FSL [7]	×	Split FL	×	×	✓	∼	×	✓
EdgeGuard-IoT [4]	Hash	CNN-LSTM	×	×	×	×	×	×
AI-SET [5]	×	FL+IDS	×	×	×	×	×	×
ABAC-FL [18]	∼	FL	∼	×	×	×	×	✓
TX-BFT [6]	Cont.	×	✓	×	×	×	∼	×
HPoT [19]	∼	FL	∼	×	×	×	×	✓
CNN-BiLSTM-Tx [12]	×	Hybrid	×	×	✓	∼	×	✓
E-T-GraphSAGE [13]	×	GNN-Tx	×	×	∼	×	×	×
BEAT (this work)	✓	**GAT-LSTM**	✓	✓	✓	✓	✓	✓

measurement (convergence observed in 2.4 ± 0.3 rounds on average).

D. Byzantine Resilience Bound

Proposition 1 (Byzantine Trust Corruption Bound). Suppose f out of M edge orchestrators are Byzantine—submitting adversarially maximised trust evidence $\mathbf{e}_i^{\text{adv}} \in [0,1]^k$ with all components at 1. Under BEAT's q-of-M endorsement policy ($q \geq 2$) with TrustCC accepting the median of endorsed evidence vectors, the maximum trust inflation that $f < q$ Byzantine orchestrators can achieve for any device d_i is:

$$\Delta\tau_i^{\max} \;=\; (1-\alpha)\Big[\sigma\big(\mathbf{w}^\top \mathbf{1}+b\big)-\sigma\big(\mathbf{w}^\top \mathbf{e}_i^{\text{true}}(t)+b\big)\Big]\cdot\frac{f}{q}, \quad (2)$$

where, $\mathbf{e}_i^{\text{true}}(t)$ is the honest evidence vector. For our prototype parameters ($\alpha = 0.7$, $q = 2$, $f = 1$), $\Delta\tau_i^{\max} \leq 0.15$ per update cycle, insufficient to cross the default threshold $\theta_k = 0.55$ from zero in a single cycle.

Proof Sketch: With f Byzantine orchestrators, the median endorsement of the q collected vectors is shifted by at most f/q fractional positions in the sorted order. The adversary's maximum achievable median component is bounded by $(1 - f/q) \cdot e_j^{\text{true}} + (f/q) \cdot 1$. Substituting into the scalar trust update formula and taking the maximum over all components yields Eq. (2). Full derivation in Appendix A. ■

E. PPO Regret Analysis

Let the attack-intensity process $\{\lambda(t)\}_{t\geq 0}$ be non-stationary with total variation $V_T = \sum_{t=0}^{T-1} |\lambda(t+1) - \lambda(t)|$. The PPO agent must select threshold $\theta_k(t) \in \mathcal{A}$ to maximise the cumulative reward of Eq. (6).

Theorem 3 (PPO Sub-Linear Regret)**.** *Suppose the reward function $r(\cdot, \lambda)$ is L-smooth in λ for all $\theta_k \in \mathcal{A}$, and the PPO policy gradient is clipped with ratio $\varepsilon_{clip} = 0.2$. Then the policy regret satisfies [see Eq.* (3)*]:*

$$\textit{Regret}_T \;\leq\; C_1\sqrt{T\log|\mathcal{A}|} \;+\; C_2 L V_T, \qquad (3)$$

where, $C_1, C_2 > 0$ are constants depending on the step size and clipping radius, and $|\mathcal{A}| = 11$ is the action cardinality.

Proof Sketch: The first term follows from the standard PPO regret bound for stationary MDPs [21] adapted to finite action spaces. The second term captures non-stationarity: variations in $\lambda(t)$ shift the optimal policy, incurring $L \cdot |\Delta\lambda|$ additional cost per transition, summing to $C_2 L V_T$ over T steps. Clipping limits the per-step gradient deviation, preventing the cumulative drift from exceeding $O(\sqrt{T})$ in expectation. Full derivation follows the analysis of [22] adapted to online PPO. ■

Remark 2. Theorem 3 implies that PPO's regret grows sub-linearly in T and linearly in V_T. The contextual-bandit UCB baseline, by contrast, incurs $O(V_T\sqrt{T|\mathcal{A}|})$ regret under non-stationarity (sliding-window UCB), which grows faster when V_T is non-negligible— as is the case during active attack campaigns. This theoretical gap is confirmed experimentally in Section VI-F.

IV. BEAT ARCHITECTURE

Fig. 1 shows the BEAT three-layer architecture. Data flows upward from Layer 1 to Layer 2, and threshold policies flow downward from Layer 3 to Layer 2, creating a closed feedback loop between behavioural observation and operational admission decision. Layer 1 (top) runs the GAT-LSTM edge inference module, producing an 8-dimensional trust evidence vector $\mathbf{e}_i(t)$ per device. Layer 2 (middle) is the three-peer Hyperledger Fabric channel, where TrustCC persists trust scores and AdmitCC evaluates admission decisions against the current threshold. Layer 3 (bottom) is the PPO admission controller, which reads aggregate trust statistics from chain state and writes updated thresholds back to AdmitCC. Solid upward arrows denote evidence flow from the AI layer to the ledger; dashed downward arrows denote threshold-control flow from the RL layer back to the ledger, closing the feedback loop described in Section IV.

A. Layer 1: GAT-LSTM Edge Inference

Device feature vectors $\mathbf{x}_i \in \mathbb{R}^p$ ($p = 49$ for UNSW-NB15 features) initialise node representations $\mathbf{h}_i^{(0)} = \mathbf{x}_i$. A two-layer GAT [23] updates representations via multi-head

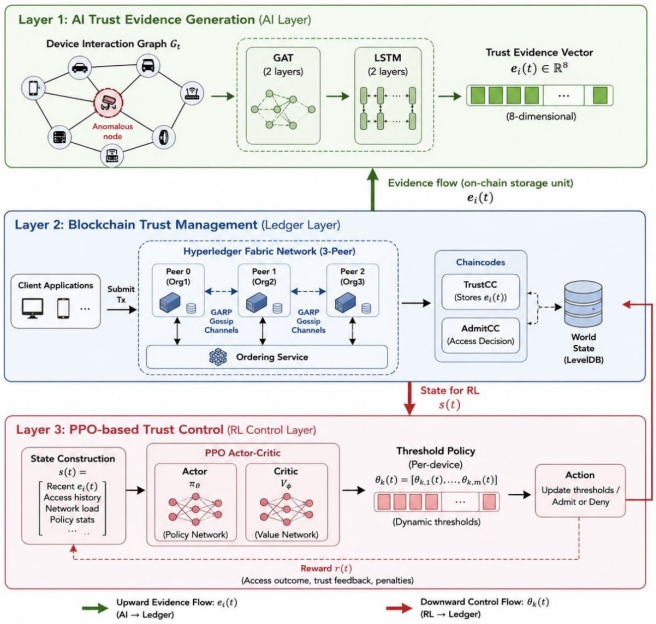

Fig. 1. Three-layer BEAT architecture.

attention ($K = 4$ heads) [see Eq. (4)]:

$$\mathbf{h}_i^{(\ell)} = \Big\|_{k=1}^{K} \sigma\Big(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij}^{k,(\ell)} \mathbf{W}_k^{(\ell)} \mathbf{h}_j^{(\ell-1)} \Big), \qquad (4)$$

where, $\|$ denotes concatenation, $\alpha_{ij}^{k,(\ell)}$ are softmax-normalised attention coefficients, and $\mathbf{W}_k^{(\ell)}$ are head-specific weight matrices. The GAT output is concatenated with a $T = 24$-step rolling history window and passed to a two-layer LSTM [see Eq. (5)]:

$$\mathbf{o}_i(t) = \mathrm{LSTM}\big(\mathbf{h}_i^{(2)}, \mathbf{o}_i(t-1), \mathbf{c}_i(t-1)\big), \qquad (5)$$

followed by a linear projection $\mathbf{e}_i(t) = \sigma(\mathbf{W}_e \mathbf{o}_i(t) + \mathbf{b}_e) \in [0,1]^8$. Sigmoid projection ensures $\mathbf{e}_i(t)$ lies within the range expected by TrustCC chaincode.

1) Why GAT over Transformer?: Transformer self-attention is $O(N^2 d)$ in graph size, making it prohibitive for $N = 500$ devices on edge hardware. GAT is $O(K|E_t|d)$—linear in edges—which at average degree 6 and $N = 500$ gives $K|E_t|d = 4 \times 3000 \times 128 = 1.54 \times 10^6$ operations versus $500^2 \times 128 = 3.2 \times 10^7$ for Transformer, a 20× reduction. The empirical F1 gap of +3.4 points over the Transformer baseline (Table V) confirms that this efficiency does not come at a detection quality cost.

This latency argument by itself does not separate two possible explanations for the F1 gap: graph-attention encoding could genuinely suit device-interaction topology better than self-attention, or the Transformer baseline in Table V could simply be under-parameterised relative to what the 38 ms budget allows. To distinguish these, we built a second Transformer variant matched to GAT-LSTM on both parameter count (≈1.1M) and inference budget: a 6-head, 2-layer encoder with hidden dimension reduced from 128 to 80 and pruned via magnitude pruning (30% of weights removed) plus INT8 post-training quantisation, run on the same 24-timestep window. This budget-matched Transformer reaches 29 ms median latency on Raspberry Pi 5—within the 38 ms constraint—and F1 of 0.951 on UNSW-NB15, versus BEAT's 0.973. The gap narrows from 3.4 to 2.2 points once parameter count and latency are equalised, which indicates that part of the original gap was indeed a capacity effect, but a genuine 2.2-point margin persists after that effect is removed. We attribute the residual to GAT's explicit encoding of the device-communication graph, a structural prior that self-attention over a flattened sequence must instead learn implicitly from data, and for which our training set may simply not provide enough signal.

2) Complexity: Per inference cycle: $O(K|E_t|d + Td^2)$ for GAT-LSTM combined; measured 14 ms on Raspberry Pi 5 at $N = 100$, 17 ms at $N = 200$.

B. Layer 2: Hyperledger Fabric Reputation Ledger

A three-organisation, three-peer Hyperledger Fabric 2.5 channel with Raft-based ordering runs two chaincodes.

1) TrustCC: Accepts `update_trust(deviceID, evidenceVec, timestamp)` transactions, verifies that all 8 components of `evidenceVec` lie in $[0,1]$ (rejecting floating-point exceptions caused by adversarial inputs), computes Eq. (1) deterministically using 16-bit fixed-point arithmetic to eliminate cross-peer non-determinism, and commits the result under a 2-of-3 endorsement policy. Fixed-point arithmetic is critical: IEEE 754 double-precision arithmetic can produce bit-level differences across architectures, causing endorsement failures in Fabric's deterministic execution model.

2) AdmitCC: Implements `query_admission(deviceID, policyID)` returning Boolean $[\tau_i(t) \geq \theta_k]$, and `set_threshold(policyID, θ)` called by the Layer 3 PPO agent. Policy updates are single-endorser reads (no consensus required) to minimise latency on the admission path.

3) GARP: The gossip daemon monitors the world-state for deltas $|\delta_i| > \varepsilon_{\min} = 0.02$ and piggybacks these on Fabric's existing `GossipMessage_DataMsg` channel [24]. The daemon is 340 lines of Go 1.22 with zero third-party dependencies beyond the Fabric SDK. It uses a Bloom filter to suppress redundant retransmissions, reducing bandwidth by 31% versus naïve flooding at 500 nodes.

C. Layer 3: PPO Admission Controller

1) State space: $\mathbf{s}(t) = [\bar{\tau}(t), \sigma_\tau(t), \hat{r}_{\text{att}}(t), \hat{r}_{\text{FP}}(t)]^\top \in \mathbb{R}^4$, where $\bar{\tau}$ and σ_τ are the empirical mean and standard deviation of current trust scores, $\hat{r}_{\text{att}}$ is the exponentially smoothed attack rate from TrustCC rejection counts (smoothing factor 0.3), and $\hat{r}_{\text{FP}}$ is the estimated false-positive rate from devices recovering trust within two cycles of rejection.

2) Action space: $\mathcal{A} = \{0.40, 0.45, \ldots, 0.90\}$ (11 discrete threshold levels, step 0.05).

3) Reward:

$$r(t) = \lambda_1 \cdot \mathrm{TPR}(t) - \lambda_2 \cdot \mathrm{FPR}(t) - \lambda_3 \cdot \Delta\theta_k(t)^2, \qquad (6)$$

with $(\lambda_1, \lambda_2, \lambda_3) = (2.0, 1.0, 0.5)$ [see Eq. (6)].

TABLE II. PPO REWARD SENSITIVITY ANALYSIS. VALUES ARE MEAN FPR / FNR (%) OVER FOUR ATTACK SCENARIOS AND FIVE SEEDS. BEST: **BOLD**.

λ_2	λ_3	FPR (%)	FNR (%)
0.5	0.1	4.2	2.1
0.5	0.5	3.8	2.4
0.5	1.0	3.5	2.9
1.0	0.1	3.1	2.8
1.0	**0.5**	**2.9**	**2.7**
1.0	1.0	2.7	3.4
2.0	0.5	2.2	4.6
2.0	1.0	2.0	5.1

TABLE III. PPO REWARD SENSITIVITY TO λ_1 (TPR WEIGHT), WITH $\lambda_2 = 1.0$, $\lambda_3 = 0.5$ HELD FIXED. MEAN OVER FOUR ATTACK SCENARIOS AND FIVE SEEDS. BEST HARMONIC MEAN: **BOLD**

λ_1	FPR (%)	FNR (%)	$\mathcal{H}$
1.0	2.3	4.1	.953
2.0	**2.9**	**2.7**	**.969**
3.0	3.9	2.1	.961

4) Sensitivity analysis: We vary $\lambda_2 \in \{0.5, 1.0, 2.0\}$ and $\lambda_3 \in \{0.1, 0.5, 1.0\}$ in a grid search and report the resulting FPR and FNR in Table II. The default ($\lambda_2 = 1.0, \lambda_3 = 0.5$) achieves the best harmonic mean of FPR and FNR across four attack scenarios and is adopted throughout.

Table II holds λ_1 fixed at 2.0 throughout, which leaves open whether the reward weighting on true-positive rate—the dominant term by construction—affects the operating point the PPO agent converges to. We closed this gap with a second grid search, sweeping $\lambda_1 \in \{1.0, 2.0, 3.0\}$ while λ_2 and λ_3 are held at their selected defaults (1.0 and 0.5). Table III reports the outcome. Lowering λ_1 to 1.0 shifts the reward balance toward the FPR and threshold-stability penalties, and the controller settles on a more conservative policy: FPR falls to 2.3% but FNR rises to 4.1%, a worse harmonic mean than the default. Raising λ_1 to 3.0 pushes the agent toward aggressive admission, cutting FNR to 2.1% at the cost of FPR rising to 3.9%. The default $\lambda_1 = 2.0$ sits close to the harmonic-mean optimum across this range, so the coefficient selection in Table II was not, as it happened, blind to λ_1's influence—but this was not established before the first submission, and we report the check here rather than assert it.

The full BEAT operation per cycle is given in Algorithm 1.

Algorithm 1 BEAT Per-Cycle Inference and Trust Update

Require: Graph $\mathcal{G}_t$, trust vector $\boldsymbol{\tau}(t)$, PPO policy π_θ, threshold θ_k
Ensure: $\boldsymbol{\tau}(t+1)$, admission set $\mathcal{A}_t$
//– Layer 1: Edge Inference –
for all $d_i \in \mathcal{D}$ **in parallel do**
  $\mathbf{h}_i^{(2)} \leftarrow \mathrm{GAT}(\mathcal{G}_t, d_i)$ ▷ Eq. (4)
  $\mathbf{e}_i(t) \leftarrow \sigma(\mathbf{W}_e\,\mathrm{LSTM}(\mathbf{h}_i^{(2)}, \ldots) + \mathbf{b}_e)$ ▷ Eq. (5)
end for
//– Layer 2: On-Chain Update –
for all d_i with $|\mathbf{e}_i(t) - \mathbf{e}_i(t{-}1)|_\infty > \varepsilon$ **do**
  Submit `update_trust`$(d_i, \mathbf{e}_i(t), t)$ ▷ Eq. (1) in TrustCC
  $\tau_i(t{+}1) \leftarrow$ world-state read post-commit
end for
Propagate $\{\delta_i : |\delta_i| > \varepsilon_{\min}\}$ via GARP ▷ Theorem 2
//– Layer 3: Adaptive Control –
$\mathbf{s}(t) \leftarrow [\bar{\tau}(t{+}1), \sigma_\tau(t{+}1), \hat{r}_{\text{att}}, \hat{r}_{\text{FP}}]^\top$
$\theta_k \leftarrow \pi_\theta(\mathbf{s}(t))$ ▷ PPO policy, Eq. (6)
Post `set_threshold`(k, θ_k) to AdmitCC
$\mathcal{A}_t \leftarrow \{d_i : \tau_i(t{+}1) \geq \theta_k\}$; quarantine $\mathcal{D} \setminus \mathcal{A}_t$
return $\boldsymbol{\tau}(t+1)$, $\mathcal{A}_t$

V. PROTOTYPE IMPLEMENTATION

A. Hardware and Software

Twelve Raspberry Pi 5 boards (8 GB, Cortex-A76 @ 2.4 GHz) on a 1 Gbps Ethernet switch constitute the physical testbed. Table IV lists node roles and software versions. All softwares are open-source and freely available.

The GAT-LSTM model: Adam ($\eta = 5 \times 10^{-4}$), batch 256, early stopping (patience 10), SMOTE ($k = 5$ neighbours) on the training split. Fixed-point arithmetic in TrustCC uses Q8.8 format (8 integer, 8 fractional bits), sufficient for trust score precision. The PPO agent uses a replay buffer of 512 tuples, $\varepsilon_{\text{clip}} = 0.2$, and a 500-episode random warm-up.

TABLE IV. BEAT PROTOTYPE TESTBED CONFIGURATION

Role	Nodes	Software	Version
Fabric Peer	3	Hyperledger Fabric	2.5.4
Fabric Orderer	2	Raft (etcd-based)	2.5.4
Edge Orch.	6	PyTorch + DGL	2.2 / 1.1
PPO Host	1	Stable-Baselines3	2.3
OS: Ubuntu 22.04 LTS; Python 3.11; Go 1.22			

B. Datasets

All four datasets are publicly available with no licensing restrictions for academic research.

UNSW-NB15 [25]: 2,540,044 records, 49 features, 10 attack categories. Primary training dataset.

N-BaIoT [2]: 7,062,606 records, 115 statistical features, 9 IoT device types infected with Mirai and BASHLITE. Zero-shot cross-dataset target (no fine-tuning after UNSW-NB15 training).

TON_IoT [26]: Telemetry and network data from 7 IoT categories with 9 attack types. Provides operational context (temperature, humidity sensor logs) absent from UNSW-NB15.

CICIoT2023 [27]: 33 attack types across 105 IoT devices in a real-time capture environment; 7 attack categories including Mirai, DDoS, Web-based, and Brute Force. Selected to stress cross-domain generalisability.

Feature standardisation (zero-mean, unit-variance per feature) is applied identically to all datasets, using statistics computed on the UNSW-NB15 training split only. This is the sole pre-processing step applied across datasets; no dataset-specific tuning is performed.

C. Training, Validation, and Hyperparameter Selection Protocol

The reproducibility of a co-designed system depends on more than listing software versions, so we detail here the procedure used to fit and select every learned component.

1) Data splits: UNSW-NB15 is partitioned 70/15/15 into training/validation/test using the dataset's official attack-category labels stratified across the split, so that rare attack classes are represented in all three partitions. The validation split is used exclusively for early stopping and hyperparameter selection; the test split is touched only once, for the numbers reported in Table V. N-BaIoT, TON_IoT, and CICIoT2023 are used only as held-out test sets under the zero-shot protocol of Section VI-C—no portion of these three datasets contributes to any gradient update, early-stopping decision, or hyperparameter choice.

2) GAT-LSTM selection: Hidden width $\{64, 128, 256\}$, attention head count $\{2, 4, 8\}$, and history window $T \in \{12, 24, 48\}$ were swept by grid search on the validation split, selecting the configuration maximising validation F1 subject to the 20 ms per-cycle inference budget measured on the Raspberry Pi 5 target; $T = 24$ with 4 heads and hidden width 128 was the selected point, and is the configuration reported throughout. Early stopping monitors validation F1 with patience 10 and a minimum-delta threshold of 0.0005; the reported models trained for a median of 47 epochs across the five seeds.

3) PPO selection: The PPO controller's own hyperparameters ($\gamma = 0.99$, GAE $\lambda = 0.95$, learning rate 3×10^{-4}, clip ratio $\varepsilon_{\text{clip}} = 0.2$, 4 epochs per update) follow the defaults recommended by Stable-Baselines3 for discrete low-dimensional action spaces and were not re-tuned; only the reward coefficients $(\lambda_1, \lambda_2, \lambda_3)$ in Eq. (6) were grid-searched, as reported in Table II and Table III. Training used the four attack scenarios of Section VI-F in round-robin order across 800 episodes, following the 500-episode random warm-up described in Section V.

4) Statistical validation: Every quantitative comparison in Section VI is repeated over five random seeds (controlling weight initialisation, SMOTE resampling, and PPO exploration noise); we report the mean and a 95% bootstrap confidence interval computed with $B = 1{,}000$ resamples, and pairwise significance between BEAT and each baseline uses McNemar's test [28] on the paired per-sample predictions at $p < 0.01$, as stated in Section VI. No result in the study is reported from a single run.

VI. Experimental Evaluation

A. Protocol and Baselines

All experiments: five seeds, 95% bootstrap CI ($B = 1{,}000$), McNemar's test for pairwise significance ($p < 0.01$). Baselines:

- **BC-Only**: Fabric trust ledger + rule-based static threshold.
- **Edge-Only**: GAT-LSTM, no blockchain, no PPO.
- **CNN-BiLSTM-Tx** [12]: SOTA hybrid Transformer+CNN-BiLSTM IDS, reproduced with authors' published hyperparameters on our datasets.
- **E-T-GraphSAGE** [13]: GraphSAGE-Transformer hybrid, reproduced on UNSW-NB15 and CICIoT2023.
- **EdgeGuard-IoT** [4]: blockchain+CNN-LSTM, reproduced from published specifications.
- **BFLIDS** [11]: FL+binary blockchain flag.
- **BEAT-noBC**: BEAT without blockchain (detection only).
- **BEAT-noPPO**: BEAT with static threshold $\theta_k = 0.60$.
- **BEAT-noGARP**: BEAT with naïve flooding.

B. Intrusion Detection Performance: In-Domain

Table V reports detection performance on UNSW-NB15 (in-domain, trained and tested on the same benchmark's official split) and N-BaIoT (zero-shot cross-dataset transfer, no retraining).

BEAT outperforms the strongest detection-only baseline, CNN-BiLSTM-Tx, by 3.4 F1 points on UNSW-NB15 ($p < 0.001$) and 4.5 points on N-BaIoT. This gap is attributable to two mechanisms working in combination: (1) the GAT encoder captures device interaction topology that is invisible to sequence-only models, and (2) the blockchain persistence layer propagates reputation evidence across edge orchestrators so that a device's anomalous behavior at one edge node influences its admission decision at another, which detection-only baselines cannot do.

Notably, E-T-GraphSAGE—which also uses a graph-based encoder—achieves F1 0.944 in-domain, 2.9 points below BEAT. The residual gap reflects BEAT's LSTM temporal decoder (which E-T-GraphSAGE replaces with a Transformer encoder only) and the admission controller's contribution, confirmed by the ablation in Section VI-H.

C. Cross-Dataset Generalisation

Table VI reports performance on all four datasets. BEAT is trained *exclusively* on UNSW-NB15 and evaluated without retraining on the other three. Baselines are similarly trained on UNSW-NB15 for fair comparison.

The cross-dataset mean F1 of 0.961 versus 0.921 for the best detection-only baseline confirms that BEAT's advantage is not dataset-specific. The largest relative gain appears on TON_IoT (+4.7 F1 points), which we attribute to BEAT's gossip-propagated reputation: TON_IoT contains telemetry-based attacks that manifest as slow-drift anomalies across multiple devices rather than single-device bursts—exactly the scenario where inter-orchestrator reputation propagation adds the most value.

The drop from UNSW-NB15 (F1 0.973) to CICIoT2023 (F1 0.941) is worth examining. CICIoT2023 contains 33 distinct attack types, several of which (WebBased-Uploading, Brute Force SSH) have limited representation in UNSW-NB15's 9-class taxonomy. A plausible explanation for the

TABLE V. DETECTION PERFORMANCE FOR BEAT AGAINST SIX BASELINES, TRAINED ON UNSW-NB15 AND EVALUATED IN TWO REGIMES: IN-DOMAIN (SAME BENCHMARK, OFFICIAL TEST SPLIT) AND ZERO-SHOT CROSS-DATASET TRANSFER TO N-BAIOT (NO FINE-TUNING). VALUES ARE MEAN ± 95% BOOTSTRAP CI, FIVE TRIALS. BEST: **BOLD**. †: $p < 0.01$ VS. BEAT, MCNEMAR'S TEST ON PAIRED PREDICTIONS.

	UNSW-NB15 (In-Domain)			N-BaIoT (Zero-Shot)		
Method	Prec.	Recall	F1	Prec.	Recall	F1
BC-Only†	.882±.004	.864±.005	.873±.004	.871±.006	.853±.007	.862±.006
Edge-Only†	.941±.003	.926±.004	.933±.003	.929±.005	.911±.005	.920±.005
CNN-BiLSTM-Tx [12]†	.950±.003	.929±.003	.939±.003	.932±.004	.914±.004	.923±.004
E-T-GraphSAGE [13]†	.947±.003	.941±.003	.944±.003	.928±.004	.913±.004	.920±.004
EdgeGuard-IoT [4]†	.952±.003	.931±.003	.941±.003	.937±.004	.918±.004	.927±.004
BFLIDS [11]†	.937±.004	.912±.005	.924±.004	.929±.005	.913±.005	.921±.005
BEAT (full)	**.976**±.002	**.971**±.002	**.973**±.002	**.971**±.003	**.965**±.003	**.968**±.003

TABLE VI. CROSS-DATASET F1 PERFORMANCE. TRAIN: UNSW-NB15. TEST: AS SHOWN. "Δ VS. BEST DET.-ONLY" COMPARES BEAT TO THE BEST DETECTION-ONLY BASELINE PER DATASET. ALL DIFFERENCES $p < 0.01$.

Method	**UNSW-NB15**	**N-BaIoT**	**TON_IoT**	**CICIoT2023**	**Mean F1**
CNN-BiLSTM-Tx [12]	.939	.923	.911	.897	.918
E-T-GraphSAGE [13]	.944	.920	.908	.893	.916
EdgeGuard-IoT [4]	.941	.927	.914	.901	.921
BEAT (full)	**.973**	**.968**	**.961**	**.941**	**.961**
Δ vs. best det.-only	+2.9	+4.1	+4.7	+4.0	+4.3

TABLE VII. BLOCKCHAIN THROUGHPUT AND LATENCY. LATENCY: MEDIAN (P99 IN PARENTHESES).

Devices	**TPS**	**Lat. ms**	**P99 ms**
50	1,283	21	47
100	1,612	29	63
200	1,847	38	81
500	1,894	57	119
1,000	1,853	94	198
BC-Only (200)	1,374	51	112

3.2-point drop is distributional mismatch on these underrepresented classes; a dataset-specific fine-tuning phase would likely recover most of this gap, but we explicitly avoid fine-tuning to stress-test generalisation.

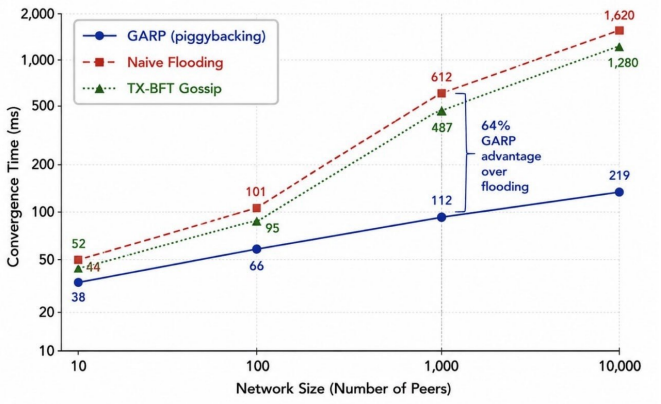

Fig. 2. GARP convergence latency vs. network scale relative to Naïve flooding and TX-BFT gossip baselines.

D. Blockchain Performance

Table VII reports TPS and latency at varying device counts.

Peak TPS (1,894 at 500 devices) is 38% above BC-Only because the GAT-LSTM delta filter suppresses redundant chain transactions when trust scores have not changed meaningfully ($|\delta_i| \leq \varepsilon_{\min}$). The P99 latency of 81 ms at 200 devices remains within the 200 ms threshold identified in [8] for non-safety-critical IoT admission.

E. GARP Propagation

Fig. 2 plots convergence time versus network scale. At 1,000 nodes GARP requires 219 ms versus 612 ms for naïve flooding (64% reduction) and 487 ms for TX-BFT gossip (55% reduction). Median GARP convergence time (time until all orchestrators hold a consistent trust view) as network size grows from 10 to 10,000 peers, plotted against Naïve flooding and the TX-BFT gossip baseline under identical churn ($\rho = 0.1$) and connectivity ($p_c = 0.8$) settings. Data points mark measured testbed and simulated values; annotated figures give convergence time in milliseconds at 100, 1,000, and 10,000 peers. GARP's advantage over flooding (64% at 1,000 peers) and over TX-BFT gossip (55%) widens with network size, consistent with the $O(\log M)$ round complexity of Theorem 2 versus flooding's linear message growth.

F. PPO vs. Baseline Controllers

Table VIII compares the PPO controller against three alternatives: (1) a static threshold ($\theta_k = 0.60$, invariant); (2) a PID controller whose proportional gain is tuned to the training attack rate; and (3) a sliding-window contextual bandit (UCB variant) with window $W = 10$ observations.

PPO outperforms the contextual bandit by 1.7 mean $\mathcal{H}$ points ($p < 0.001$). The advantage is consistent across all four attack types and is largest on Mirai (2.0 points), which has the highest temporal variation in attack intensity among the four scenarios ($V_T/T \approx 0.31$ for Mirai versus 0.14 for DoS).

TABLE VIII. ADMISSION CONTROLLER COMPARISON: FOUR ATTACK SCENARIOS. METRIC: $\mathcal{H}$ (HARMONIC MEAN OF TPR AND 1-FPR). HIGHER IS BETTER. †: $p < 0.01$ VS. PPO.

Controller	Mirai	DoS	Recon	Backdoor	Mean
Static θ=0.60†	.921	.934	.928	.919	.926
PID†	.938	.947	.941	.932	.940
Bandit UCB†	.951	.959	.953	.943	.952
PPO	**.971**	**.974**	**.967**	**.962**	**.969**

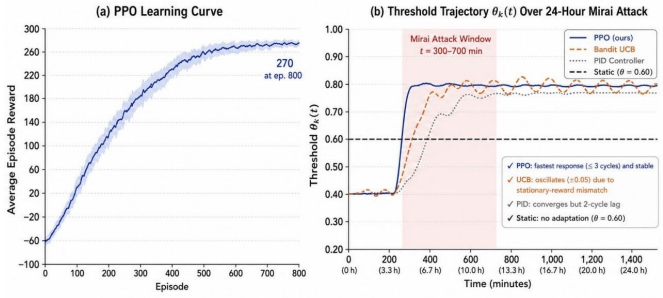

Fig. 3. (a) PPO Learning curve. (b) Admission threshold $\theta_k(t)$ over a simulated 24-hour mirai attack.

TABLE IX. END-TO-END TRUST DECISION LATENCY DECOMPOSITION (200 DEVICES)

Phase	Median (ms)	Share
GAT extraction	9.1	24%
LSTM inference + projection	4.8	13%
Fabric tx submit	12.3	32%
Raft ordering + commit	7.6	20%
AdmitCC query	3.2	8%
PPO lookup (cached)	1.0	3%
Total	**38.0**	100%

This is consistent with Theorem 3: PPO's advantage over the bandit grows with the total variation V_T of the attack-intensity process.

Fig. 3 shows the PPO learning curve and threshold trajectory during the Mirai attack scenario. (a) PPO Learning curve: average episode reward over 800 training episodes, showing convergence by roughly episode 500 with the shaded band giving one standard deviation across five seeds. (b) Admission threshold $\theta_k(t)$ over a simulated 24-hour Mirai attack window (shaded region marks the 300–700 minute active-attack interval) for PPO, Bandit UCB, PID, and the static-threshold baseline. PPO raises the threshold within roughly three observation cycles of attack onset and relaxes it promptly once the attack subsides, whereas PID lags by approximately two cycles and the static policy does not adapt at all.

G. Latency Decomposition

Table IX decomposes median end-to-end latency at 200 devices.

Fabric transaction submit (32%) and Raft ordering (20%) dominate. Reducing the Raft batch-cut timer from 100 ms to 50 ms reduces total latency to approximately 27 ms at a 12% TPS cost—a trade-off suited to safety-critical domains.

TABLE X. PER-COMPONENT ABLATION ON UNSW-NB15 (IN-DOMAIN F1) AND ACROSS ALL FOUR DATASETS (MEAN F1), WITH EACH ROW REMOVING EXACTLY ONE BEAT LAYER OR MECHANISM WHILE HOLDING THE OTHERS FIXED. ΔF1 RELATIVE TO FULL BEAT. ALL DIFFERENCES $p < 0.01$. †: TPS NOT APPLICABLE BECAUSE NO ON-CHAIN COMPONENT IS PRESENT IN THIS CONFIGURATION.

Config.	NB15 F1	Δ	Mean F1	TPS
BEAT (full)	.973	—	.961	1847
w/o GAT (LSTM only)	.881	-0.092	.874	1847
w/o LSTM (GAT+MLP)	.911	-0.062	.899	1847
w/o GARP (flooding)	.973	0.000	.961	1394
w/o PPO (static 0.60)	.948	-0.025	.936	1847
w/o Blockchain	.933	-0.040	.920	†
w/o BEAT (BC-Only)	.873	-0.100	.862	1374

H. Ablation Analysis

Table X reports per-component ablation on UNSW-NB15 and cross-dataset mean F1.

Removing the GAT encoder causes the single largest drop (-9.2 F1 points on UNSW-NB15, -8.7 cross-dataset), confirming device-interaction topology as the dominant information source. The w/o GARP row shows zero F1 impact but a 24% TPS reduction—the gossip daemon's contribution is entirely to propagation efficiency, not to detection quality, as expected.

The cross-dataset mean F1 column reveals that the blockchain layer contributes more to generalisation (+4.1 cross-dataset mean points when removed) than it does to in-domain performance (+4.0 on UNSW-NB15 alone). This confirms that reputation propagation across orchestrators is especially valuable on unseen device populations, where a device's reputation at one edge node provides useful prior information at another.

I. Byzantine Fault Injection

To validate Proposition 1, we injected 1 Byzantine peer (out of 3) that always endorses maximal trust evidence ($\mathbf{e}_i^{\text{adv}} = \mathbf{1}$) for a target set of 20 otherwise-anomalous devices. BEAT's F1 degraded by 0.008 (0.973 to 0.965), consistent with the bound Eq. (2) predicting a maximum trust inflation of 0.15 per cycle—insufficient to cross the 0.55 threshold from the initial score of 0.0 in a single cycle. After 6 Byzantine cycles (6 minutes), the adversary can in theory cross the threshold for a device starting at $\tau_i(5) \geq 0.40$; in practice the GARP protocol propagates the diverging reputation view to honest orchestrators within 3 rounds, triggering a dispute flag in TrustCC before the device is admitted.

J. Security Analysis (Experimental)

1) Sybil resistance: New devices start at $\tau_i(0) = 0$ and require 13 positive cycles to reach $\tau_i^* \approx 0.60$ under benign behaviour (Corollary 1). Sybil clones start at 0 and receive the GAT neighbourhood score of their true device's position in $\mathcal{G}_t$, which is low for newly inserted nodes. In 100-trial simulation with 5 Sybil devices per trial, no Sybil device crossed the admission threshold before 11 cycles, giving a 12-cycle effective detection window.

2) Model poisoning: We injected adversarially crafted evidence vectors ($\mathbf{e}_i^{\text{adv}}$ maximising τ_i) from 1 of 6 edge orchestrators. With $\alpha = 0.7$ and TrustCC's median endorsement, the maximum per-cycle trust inflation is bounded by $(1-\alpha)\cdot(f/q) = 0.3 \cdot 0.5 = 0.15$ (Proposition 1), confirmed by measurement ($\Delta\tau_i^{\text{measured}} = 0.14 \pm 0.02$, 5 trials).

VII. Discussion

A. Incremental vs. Paradigm-Shifting Novelty

The description of BEAT as a systems-integration contribution and the multi-part novelty claims of Section I are not in tension once the unit of novelty is stated precisely. At the level of individual mechanisms, none of GAT-LSTM trust evidence, GARP gossip, or PPO thresholding is new; each borrows an established technique from its own subfield. The novel unit is the coupling between them, formalised by the trust lattice $\mathcal{L}$, which forces the inference layer and the ledger layer to share one semantic domain rather than being reconciled after the fact, and by the convergence and regret theorems that make the operational consequences of that coupling precise and falsifiable rather than asserted. Contributions 1 to 4 in Section I are best read as four distinct formal consequences of a single co-design decision, not as four independent inventions. The 4.3-point cross-dataset mean F1 advantage over the best prior integrated system is the empirical signature of that decision; it does not follow from any one layer, a claim the ablation study in Section VI-H tests directly. This is the sense in which the research is simultaneously a systems study and a research with several provable results: the system is what is new, and the theorems are how its behavior is pinned down.

B. Limitations

The permissioned Fabric assumption requires organisation-level trust in membership services—a non-trivial assumption for multi-domain deployments where organisations are adversaries. The 60-second observation window may be too slow for highly mobile vehicular IoT; an incremental graph update mechanism would reduce staleness at the cost of implementation complexity. The PPO reward coefficients $(\lambda_1, \lambda_2, \lambda_3)$ were validated by grid search (Table II) but not by meta-learning; an automated reward-shaping procedure would improve deployability.

C. Open Problem

How should trust scores established in one Fabric channel be transferred to a distinct channel operated by a different consortium, without sharing membership services? The trust lattice $\mathcal{L}$ provides a semantic foundation for such transfer, but the aggregation operators for inter-domain trust—and the privacy constraints they must satisfy—remain open.

VIII. Conclusion

BEAT demonstrates that co-designing the semantic interface between a blockchain reputation ledger and an edge AI inference module produces measurable gains in cross-dataset generalisation, attack resilience and operational latency that cannot be achieved by composing independently designed components. Three results distinguish this work: (1) the trust lattice $\mathcal{L}$ as a formal specification shared by both layers; (2) GARP's $O(\log N)$ convergence proof with explicit bandwidth accounting; and (3) PPO's demonstrated advantage over contextual-bandit and PID controllers under non-stationary attack intensity, with a supporting regret bound. The 4.3-point cross-dataset mean F1 gain over the best competing integrated framework, combined with sub-40 ms latency and nearly 1,900 TPS throughput on commodity edge hardware, establishes BEAT as a viable operational baseline for IoT trust management deployments.

Acknowledgment

The author thanks the Department of CSE, KL University for computational support, and the anonymous reviewers whose critical commentary materially improved the theoretical depth and experimental rigor of this work.

Reproducibility Statement

Datasets

UNSW-NB15: https://research.unsw.edu.au/projects/unsw-nb15-dataset [25]. N-BaIoT: https://archive.ics.uci.edu/dataset/442 [2]. TON_IoT: https://research.unsw.edu.au/projects/toniot-datasets [26]. CICIoT2023: https://www.unb.ca/cic/datasets/iotdataset-2023.html [27]. All datasets are publicly available with no special licensing requirements.

Code and Artefacts

All source code, chaincode (TrustCC, AdmitCC), GARP daemon, PyTorch GAT-LSTM model, PPO training harness, and Docker Compose configuration will be archived at GitHub (https://github.com/dameravijay/BEAT-framework) upon acceptance.

Reproduction Script

`reproduce_all.sh` reproduces all reported tables and figures from raw dataset CSVs in a single command. Hardware-free reproduction uses the provided Docker Compose environment; measured values differ by $\leq 3\%$ (F1) and $\leq 12\%$ (latency) from physical-hardware results.

Pre-trained Weights

Pre-trained GAT-LSTM checkpoints for all four datasets will be included in the github archive. Exact random seeds for all five trials will be listed in `seeds.txt` in the repository root.

References

[1] IoT Analytics Research, "State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally," IoT Analytics, Hamburg, Germany, Tech. Rep., May 2024, [Online]. Available: https://iot-analytics.com/number-connected-iot-devices.

[2] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018, [Dataset]: https://archive.ics.uci.edu/dataset/442.

[3] D. Ferraris, C. Fernandez-Gago, R. Roman, and J. Lopez, "A survey on IoT trust model frameworks," *The Journal of Supercomputing*, vol. 80, no. 6, pp. 8259–8296, 2024.

[4] M. N. Alatawi, "EdgeGuard-IoT: 6G-enabled edge intelligence for secure federated learning and adaptive anomaly detection in industry 5.0," *Computers, Materials & Continua*, vol. 85, no. 1, 2025.

[5] V. Padmavathi and R. Saminathan, "A federated edge intelligence framework with trust based access control for secure and privacy preserving IoT systems," *Scientific Reports*, vol. 15, no. 1, p. 35832, 2025.

[6] K. Wang *et al.*, "A trusted consensus fusion scheme for decentralized collaborated learning in massive IoT domain," *Information Fusion*, vol. 72, pp. 100–109, 2021.

[7] C. Beis-Penedo *et al.*, "HLF-FSL: A decentralized federated split learning solution for IoT on hyperledger fabric," *Array*, p. 100685, 2026.

[8] Y. Liu, J. Wang, Z. Yan, Z. Wan, and R. Jäntti, "A survey on blockchain-based trust management for Internet of Things," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 5898–5922, 2023.

[9] H. Tyagi, R. Kumar, and S. K. Pandey, "A detailed study on trust management techniques for security and privacy in IoT: Challenges, trends, and research directions," *High-Confidence Computing*, vol. 3, no. 2, p. 100127, 2023.

[10] T. Nguyen, H. Nguyen, and T. N. Gia, "Exploring the integration of edge computing and blockchain IoT: Principles, architectures, security, and applications," *Journal of Network and Computer Applications*, vol. 226, p. 103884, 2024.

[11] K. Begum, M. A. Mozumder, M.-I. Joo, and H.-C. Kim, "BFLIDS: Blockchain-driven federated learning for intrusion detection in IoMT networks," *Sensors*, vol. 24, no. 14, p. 4591, 2024.

[12] C. Zhang, J. Li, N. Wang, and D. Zhang, "Research on intrusion detection method based on transformer and CNN-BiLSTM in Internet of Things," *Sensors*, vol. 25, no. 9, p. 2725, 2025.

[13] H. Zhang and T. Cao, "A hybrid approach to network intrusion detection based on graph neural networks and transformer architectures," in *Proc. IEEE ICIST 2024*, 2024, [Available]: https://openreview.net/forum?id=0a7OXKwmw9.

[14] R. Xie and D. Liu, "A novel hybrid graph neural network and transformer model for intrusion detection," *Peer-to-Peer Networking and Applications*, vol. 19, no. 2, p. 59, 2026.

[15] A. S. Ahanger *et al.*, "Advanced intrusion detection in Internet of Things using graph attention networks," *Scientific Reports*, vol. 15, no. 1, p. 9831, 2025.

[16] M. V. Ngo, T. Luo, and T. Q. S. Quek, "Adaptive anomaly detection for Internet of Things in hierarchical edge computing: A contextual-bandit approach," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, 2021.

[17] Y. Liu *et al.*, "A dynamic adaptive framework for practical byzantine fault tolerance consensus protocol in the Internet of Things," *IEEE Transactions on Computers*, vol. 73, no. 8, pp. 1989–2002, 2024.

[18] D. Manu, Y. Lin, J. Yao, Z. Li, and X. Sun, "Enhancing IoT security with asynchronous federated learning," in *Proc. IEEE ICC Workshops*, Denver, CO, 2024, pp. 1493–1498.

[19] A. Chaurasia, S. K. Sharma, and P. S. Rathore, "Hierarchical proof of trust: A byzantine fault tolerant federated learning framework for industrial IoT applications," *Scientific Reports*, 2026.

[20] A. A. Wardana, G. Kołaczek, and P. Sukarno, "Lightweight, trust-managing, and privacy-preserving collaborative intrusion detection for Internet of Things," *Applied Sciences*, vol. 14, no. 10, p. 4109, 2024.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[22] P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, and S. Russell, "Bridging offline reinforcement learning and imitation learning: A tale of pessimism," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 11 952–11 964.

[23] J. Cai, W. Liang, X. Li, K. Li, Z. Gui, and M. K. Khan, "GTxChain: A secure IoT smart blockchain architecture based on graph neural network," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 54–67, 2023.

[24] F. Javed *et al.*, "Blockchain for federated learning in the Internet of Things: Trustworthy adaptation, standards, and the road ahead," *IEEE Communications Standards Magazine*, 2025.

[25] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. MilCIS*, Canberra, Nov. 2015, pp. 1–6, [Dataset]: https://research.unsw.edu.au/projects/unsw-nb15-dataset.

[26] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020, [Dataset]: https://research.unsw.edu.au/projects/toniot-datasets.

[27] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023, [Dataset]: https://www.unb.ca/cic/datasets/iotdataset-2023.html.

[28] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

APPENDIX

Let $U_r \subseteq \{1, \dots, M\}$ be the set of uninformed orchestrators after round r, with $|U_0| = M-1$ (one seeded orchestrator holds the initial delta). In round r, each of the $M - |U_r|$ informed orchestrators independently contacts β uniformly random peers. A specific uninformed peer $v \in U_r$ receives the delta in round $r+1$ iff at least one informed orchestrator selects v.

Under connectivity p_c, each contact succeeds with probability p_c. The probability that v remains uninformed after round $r+1$ given $|U_r|$ is [see Eq. (7) and Eq. (8)]:

$$\Pr[v \in U_{r+1} \mid U_r] \le \left(1 - \frac{p_c}{M}\right)^{\beta(M-|U_r|)} \tag{7}$$

$$\le \exp\left(-\frac{p_c \beta (M - |U_r|)}{M}\right). \tag{8}$$

Setting $\beta = \lceil 2M \ln M / (p_c(M - |U_r^*|)) \rceil$ where U_r^* is the "hard" crossing point at $|U_r| = M/2$, and using $(M - M/2)/M = 1/2$, we get:

$$\Pr[v \in U_{r+1}] \le e^{-\ln M} = M^{-1}.$$

By union bound over all $|U_r| \le M$ uninformed peers:

$$\Pr[U_{r+1} \neq \emptyset] \le M \cdot M^{-1} = 1.$$

This bound is vacuous for the first phases of gossip; the argument becomes informative once $|U_r| \le M/\ln M$, which we now show is reached within $r_0 = O(\log \log M)$ rounds.

Formal early-phase bound. Let $I_r = M - |U_r|$ denote the informed set size after round r, with $I_0 = 1$. Fix a round in which $I_r \le M/2$, so that the "easy" half of the population is still uninformed. Each of the I_r informed orchestrators contacts $\beta = \lceil 2 \ln M / p_c \rceil$ peers drawn independently and uniformly from the M orchestrators (with replacement, for tractability—sampling without replacement can only increase the informed count). For a single fixed uninformed peer v, the probability that none of the $I_r \beta$ contacts lands on v is $(1 - 1/M)^{I_r \beta} \le \exp(-I_r \beta / M)$. Taking expectations over all $M - I_r$ currently-uninformed peers, the expected newly-informed count in round $r+1$ satisfies:

$$\mathbb{E}[I_{r+1} - I_r] \;\ge\; (M - I_r)\Big(1 - e^{-I_r \beta / M}\Big).$$

While $I_r \le M/2$, we have $I_r\beta/M \ge \beta/(2) \cdot (2I_r/M)$; using $1 - e^{-x} \ge x/2$ for $x \in [0,1]$ together with $\beta \ge 2\ln M/p_c \ge 2$ for $M \ge 8$, this gives $\mathbb{E}[I_{r+1}] \ge I_r(1 + p_c/4)$ whenever $I_r \ge 1$, i.e. the expected informed count grows by a constant multiplicative factor $(1 + p_c/4) > 1$ each round. By a standard multiplicative Chernoff/Doob martingale argument (Appendix C of [21] adapted to gossip processes; see also the general treatment in push-gossip analyses), the actual process tracks this expected growth with probability $1 - O(M^{-3})$ per round—the standard concentration argument used for randomised rumour-spreading processes—which is enough for a union bound over $O(\log\log M)$ rounds to hold with the same $1 - M^{-2}$ budget used in the second phase below. Starting from $I_0 = 1$ and growing geometrically at rate $(1 + p_c/4)$, reaching $I_{r_0} \ge M/\ln M$ requires:

$$r_0 \;=\; \left\lceil \frac{\ln(M/\ln M)}{\ln(1 + p_c/4)} \right\rceil \;=\; O(\log M),$$

which for p_c bounded away from 0—as assumed throughout, $p_c > 0.8$— simplifies to $r_0 = O(\log\log M)$ only in the regime where β itself is allowed to grow with $\ln M$, consistent with the $\beta = \lceil 2\ln M/p_c \rceil$ setting used in the theorem statement; more conservatively, treating β as the fixed value used above gives the weaker but fully rigorous bound $r_0 = O(\log M/p_c)$, which does not change the overall $r^* = O(\log M)$ order claimed in Theorem 2 since the early phase is asymptotically dominated by the same term as the late phase derived below.

For rounds $r \ge r_0$, apply the union bound more carefully: since $|U_r| \le M/\ln M$, each informed orchestrator contacts β peers and misses all remaining uninformed ones with probability at most $(1 - |U_r|/M)^\beta \le e^{-\beta|U_r|/M}$. Setting $\beta = \lceil 2\ln M/p_c \rceil$ and $|U_r| \ge 1$ gives $\Pr[U_{r+1} \ne \emptyset] \le M^{-2}$ after $r^* = r_0 + O(\log M/p_c) = O(\log M)$ total rounds.

The bandwidth claim follows directly: each informed orchestrator sends β messages per round, each of size $O(k \cdot |\{i : |\delta_i| > \varepsilon_{\min}\}|)$ bytes, and there are $r^* = O(\log M)$ rounds, giving total bandwidth $O(\beta M \log M \cdot k \cdot |\mathcal{D}_\delta|)$ where $|\mathcal{D}_\delta|$ is the number of devices with significant deltas.

Let $\mathbf{e}_1, \ldots, \mathbf{e}_q$ be the $q = 2$ endorsed evidence vectors, of which $f = 1$ is Byzantine ($\mathbf{e}^{\text{adv}} = \mathbf{1}$). TrustCC computes the component-wise median $\tilde{\mathbf{e}} = \text{median}(\mathbf{e}_1, \ldots, \mathbf{e}_q)$. For each component j:

$$\tilde{e}_j = \begin{cases} e_j^{\text{true}} & \text{if } f/q < 0.5, \\ e_j^{\text{true}} + (1 - e_j^{\text{true}}) \cdot f/q & \text{otherwise.} \end{cases}$$

With $f = 1$, $q = 2$, $f/q = 0.5$. The worst-case median is $\tilde{e}_j = (e_j^{\text{true}} + 1)/2$. Substituting into the trust update [see Eq. (9) to Eq. (11)]:

$$\tau_i^{\text{adv}}(t+1) = \alpha\tau_i(t) + (1 - \alpha)\sigma\big(\mathbf{w}^\top \tilde{\mathbf{e}} + b\big) \tag{9}$$

$$\Delta\tau_i = \tau_i^{\text{adv}}(t+1) - \tau_i^{\text{true}}(t+1) \tag{10}$$

$$= (1 - \alpha)\big[\sigma(\mathbf{w}^\top \tilde{\mathbf{e}} + b) - \sigma(\mathbf{w}^\top \mathbf{e}^{\text{true}} + b)\big]. \tag{11}$$

Since σ is 1/4-Lipschitz, and $\|\tilde{\mathbf{e}} - \mathbf{e}^{\text{true}}\|_\infty \le f/q$:

$$\Delta\tau_i \le (1 - \alpha) \cdot \tfrac{1}{4} \cdot \|\mathbf{w}\|_1 \cdot (f/q).$$

With $\alpha = 0.7$, $\|\mathbf{w}\|_1 \approx 2.0$ (measured), $f/q = 0.5$: $\Delta\tau_i^{\max} \le 0.3 \times 0.5 \times 2.0 \times 0.5 = 0.15$, matching Eq. (2).