Humanize before you submit

Likely Turnitin score: **40-75%** - over 20%

Humanize the colored text

Keep humanizing until this report says Ready to submit.

Once the title says Ready to submit, you're covered: if Turnitin flags it after that, get a free month of Pro Bundle ($14.99) - plus $25 if the text came straight from our humanizer. This report isn't covered yet. See terms

Reads as AI - catches 83 of 100 sentences Turnitin flags

Borderline - with blue, catches 95 of 100; marks a bit more than Turnitin

Measured on 1,707 Turnitin reports of 20% or more, October 2026. Papers vary.

Highlights are our estimate - Turnitin may mark other sentences.

EliteWorks is not affiliated with Turnitin.

Abstract—Directed Acyclic Graph (DAG)-based distributed ledgers offer parallel transaction processing suited to IoT environments, but their tip selection algorithms (TSAs) face a fundamental tension between scalability and security. The widely used Markov Chain Monte Carlo (MCMC) algorithm relies on a static parameter that cannot simultaneously optimize for low tip accumulation and resistance to parasite chain attacks. This paper introduces the Context-Aware Adaptive Tip Selection (CATS) algorithm, which resolves this trade-off through three modules: dynamic bias adaptation driven by real-time topological signals, statistical anomaly detection with adaptive thresholding, and fairness-preserving tip reintegration. We validate CATS through discrete-event simulations against MCMC, URTS, S-URTS, and G-IOTA across varying arrival rates and attack scenarios. Results demonstrate that CATS achieves tip count stability within 5% of URTS while reducing parasite chain tip selection probability to below 0.001 across all tested chain lengths, and provides a recommended reference configuration for practical deployment.

Index Terms—directed acyclic graph, tip selection algorithm, IOTA Tangle, consensus mechanism, parasite chain, adaptive security, Internet of Things

I. Introduction

The proliferation of Internet of Things (IoT) devices demands distributed ledger technologies (DLTs) capable of processing feeless, high-throughput transactions. Although blockchain architectures pioneered decentralized consensus, their sequential block structure limits throughput and imposes fees that are unsuitable for devices limited in resources [1], [2]. DAG-based ledgers, such as the IOTA Tangle [3], address these limitations by allowing asynchronous parallel transaction attachment, where each new transaction approves prior unapproved transactions called *tips* [4].

The tip selection algorithm (TSA) that governs this attachment process is central to the operation of a DAG ledger, simultaneously controlling scalability (tip count, confirmation latency), liveness (probability that honest transactions are confirmed), and security (resistance to double-spending via parasite chains) [3], [5]. However, these systems remain subject to a fundamental tension between scalability and security [6]. The canonical MCMC algorithm uses a biased random walk parameterized by α: high α favors security but orphans honest tips; low α ensures coverage but permits adversarial tip selection [3], [7].

Despite these advances, no existing TSA simultaneously achieves: (i) dynamic response to changing network conditions; (ii) statistically principled anomaly detection that adapts to non-stationary arrival rates; and (iii) fairness guarantees that prevent false-positive quarantine of honest tips. This paper introduces the Context-Aware Adaptive Tip Selection (CATS) algorithm addressing all three requirements. We present the algorithm's formal specification, validate it through discrete-event simulation, and provide a reference parameter configuration for practical deployment.

II. Background

A. DAG Model and Tip Selection

Let $G(t) = (V(t), E(t))$ denote the DAG at time t. A tip is a transaction with in-degree zero: $L(t) = \{v \in V(t) : \text{in-deg}(v) = 0\}$. Each new transaction selects $m = 2$ tips to approve. The cumulative weight of transaction i is $H_i(t) = 1 + |\{z \in V(t) : z \text{ approves } i\}|$. The MCMC walk transitions from site j to neighbor k with probability

$$p_{jk} = \frac{\exp(-\alpha \cdot (H_j - H_k))}{\sum_z \exp(-\alpha \cdot (H_j - H_z))} \quad (1)$$

where α governs walk determinism [3].

B. Adversarial Model

A parasite chain attack [3], [8] involves an adversary secretly constructing a sub-DAG referencing a double-spending transaction. The attacker broadcasts this chain after the honest transaction has been acted upon, attempting to redirect future approvals toward the conflicting transaction. A simple parasite chain (SPC) is a linear sequence of m attacker transactions: only the first attacker transaction references the main DAG at the attachment point, and each subsequent transaction references only its predecessor within the chain [8]. The attacker issues transactions at a rate $\mu < \lambda$ (honest majority assumption). Analysis shows p_2, the probability that a tip selection algorithm selects an SPC tip, exhibits non-monotone dependence on α: no static value universally minimizes attack success [8], [7], [6].

Beyond SPCs, we consider lazy tips (transactions approving already-confirmed transactions [5]), Sybil-augmented parasite chains (where multiple fake identities generate the adversarial sub-DAG [9]), and adaptive adversaries that attempt to manipulate topological signals to keep CATS in a low-security state. Eclipse attacks [10], while primarily a network-layer

concern, affect the local DAG view on which any TSA operates.

III. The CATS Algorithm

CATS integrates three modules executed for each new transaction. Let $L^* = 2\lambda h$ denote the theoretical URTS steady-state tip count [3], where h is the proof-of-work delay. We define the tip count ratio $\rho(t) = |L(t)|/L^*$ and the weight dispersion $\delta(t) = \sigma_H/\mu_H$, where μ_H and σ_H are the mean and standard deviation of cumulative weights across tips. The set $Q(t)$ denotes tips currently held in quarantine by Module 2.

A. Module 1: Dynamic α-Adaptation

A composite state signal $S(t) \in [0,1]$ synthesizes three indicators. The *tip saturation signal* $\sigma_1 = \text{clamp}((\rho(t) - 1)/(\rho_{\max} - 1), 0, 1)$ measures tip accumulation relative to steady state, where $\rho_{\max}$ is the ratio above which tip growth is considered anomalous. The *weight divergence signal* $\sigma_2 = \min(1, \delta(t)/\delta_{\max})$ captures structural heterogeneity in cumulative weights, where $\delta_{\max}$ is a normalization constant. The *anomaly density signal* $\sigma_3 = |Q(t)|/|L(t)|$ reflects current quarantine pressure. The combined signal and adaptation rule are:

$$S(t) = w_1\sigma_1 + w_2\sigma_2 + w_3\sigma_3 \tag{2}$$

$$\alpha(t) = \alpha_{\min} + S(t) \cdot (\alpha_{\max} - \alpha_{\min}) \tag{3}$$

$$\tilde{\alpha}(t) = \gamma\,\alpha(t) + (1 - \gamma)\tilde{\alpha}(t - 1) \tag{4}$$

where $\gamma \in (0,1]$ is the exponential moving average smoothing factor. Under benign conditions $S(t) \approx 0$, yielding $\tilde{\alpha} \approx \alpha_{\min}$; under adversarial pressure the signals elevate $\tilde{\alpha}$ toward $\alpha_{\max}$.

B. Module 2: Adaptive Anomaly Detection

We construct an absorbing Markov chain from $G(t)$ following [5]: tips become absorbing states, edges are reversed, and the stationary distribution $D(t)$ is computed via power iteration with the current $\tilde{\alpha}(t)$. The adaptive threshold is:

$$T(t) = \max\big(\mu_D(t) - \kappa \cdot \sigma_D(t),\ \varepsilon\big) \tag{5}$$

where μ_D and σ_D are the mean and standard deviation of $D(t)$, κ controls detection sensitivity, and $\varepsilon > 0$ is a small floor ensuring that tips with zero selection probability (unreachable from the main DAG) are always detected. Tips with $D_i(t) < T(t)$ are quarantined into $Q(t)$. A secondary structural check classifies quarantined tips as *strongly anomalous* (chain-like subgraph, branching factor ≤ 1.2) or *weakly anomalous* (normal branching).

Algorithm 1 CATS Tip Selection

Require: $G(t), V(t), E(t), \tilde{\alpha}(t-1), Q(t-1)$

Ensure: $\text{tip}_1, \text{tip}_2$

Compute $\rho(t)$, $\delta(t)$, $|Q(t-1)|/|L(t)|$
$S(t) \leftarrow w_1\sigma_1 + w_2\sigma_2 + w_3\sigma_3$
$\alpha(t) \leftarrow \alpha_{\min} + S(t)(\alpha_{\max} - \alpha_{\min})$
$\tilde{\alpha}(t) \leftarrow \gamma\,\alpha(t) + (1 - \gamma)\tilde{\alpha}(t - 1)$
Build absorbing Markov chain with $\tilde{\alpha}(t)$
Compute $D(t)$ via power iteration
$T(t) \leftarrow \max(\mu_D(t) - \kappa \cdot \sigma_D(t), \varepsilon)$
for each tip $v_i \in L(t)$ do
    if $D_i(t) < T(t)$ then
        Add v_i to $Q(t)$; classify strong/weak
    end if
end for
if reintegration interval elapsed then
    Re-evaluate weakly anomalous tips in $Q(t)$
end if
$L'(t) \leftarrow L(t) \setminus Q(t)$
$\text{tip}_1, \text{tip}_2 \leftarrow$ uniform random select from $L'(t)$
return $\text{tip}_1, \text{tip}_2$

C. Module 3: Fairness-Preserving Reintegration

Weakly anomalous tips are periodically re-evaluated (every Δt_r transactions). If $D_i'(t) \geq T(t)$ upon recomputation, the tip is restored to $L'(t)$. Tips exceeding maximum quarantine age $t_{\max}$ are released with reduced priority. Strongly anomalous tips remain permanently excluded. The final selection applies URTS to $L'(t) = L(t) \setminus Q(t)$.

D. Integrated Execution Flow

The complete CATS procedure is summarized in Algorithm 1.

IV. Simulation and Evaluation

A. Experimental Setup

We implemented a discrete-event Tangle simulator following the agent-based model of [8] and [11]. Transactions arrive as a Poisson process with rate λ. To model the proof-of-work (PoW) delay, all transactions arriving within a window of duration $h = 1$ s observe the same DAG snapshot and select tips independently; this produces the characteristic DAG width proportional to λh, consistent with the theoretical steady-state tip count $L^* = 2\lambda h$ derived in [3]. Each experiment generates a tangle of $N = 2000$ transactions with a subtangle computation window of 500. We evaluate across $\lambda \in \{5,10,15,20\}$ transactions/second and compare CATS against five baselines:

URTS, MCMC with $\alpha = 0.001$ (MCMC$_1$), MCMC with $\alpha = 0.05$ (MCMC$_5$), S-URTS [5], and G-IOTA [12]. For security evaluation, simple parasite chains (SPCs) of lengths $m \in \{10,30,50\}$ are attached at the midpoint of the tangle. Each SPC is a linear chain: only the first attacker transaction references the main DAG at the attachment point, and every subsequent transaction references only its predecessor within

TABLE I
CATS REFERENCE CONFIGURATION

Parameter	Value	Rationale
α_{min}	0.001	Optimal for detection [5]
α_{max}	0.05	Effective SPC deterrence [8]
w_1,w_2,w_3	0.3, 0.4, 0.3	Weight divergence most sensitive
γ	0.15	Prevents oscillation
κ	1.5	<2% false positive rate
ρ_{max}	3.0	3× steady-state = anomaly
Δt_r	50 txns	Balance detection and cost
t_{max}	200 txns	Allow honest tip recovery

TABLE II MEAN TIP COUNT
(BENIGN CONDITIONS, N =2000)

λ	URTS	MCMC$_1$	MCMC$_5$	S-URTS	G-IOTA	CATS
5	5.9	6.2	6.2	5.8	4.9	6.2
10	11.7	12.7	12.7	13.3	10.1	12.2
15	17.1	20.0	19.5	17.9	15.7	17.3
20	23.1	28.4	29.2	23.1	21.0	23.0

the chain, following the model in [8]. The attacker operates at rate $\mu = 5$ (25% of network power at $\lambda = 15$), consistent with the honest majority assumption $\mu < \lambda$. Each configuration is averaged over 100 independent runs. All algorithms were implemented in Python 3.11.

B. Reference Configuration

Based on parameter sensitivity analysis, we provide the reference configuration shown in Table I. The weights $w_1 = 0.3$, $w_2 = 0.4$, $w_3 = 0.3$ reflect the empirical finding that weight divergence (σ_2) is the most sensitive early indicator of parasite chain presence, while tip saturation (σ_1) and anomaly density (σ_3) provide complementary signals. The sensitivity parameter $\kappa = 1.5$ yields less than 2% false positive rate in our experiments.

C. Scalability Results

Table II summarizes tip count behavior across arrival rates. Under benign conditions (no attack), the mean tip count for CATS across all tested λ values is within 5% of URTS, confirming that the adaptive mechanism imposes no significant scalability penalty during normal operation. Both MCMC$_1$ and MCMC$_5$ produce consistently more tips than URTS, with MCMC$_5$ reaching 1.26× the URTS count at $\lambda = 20$ due to high-α orphaning. S-URTS tracks URTS closely at low arrival rates but shows moderate divergence at $\lambda = 10$. GIOTA achieves the lowest tip count across all tested λ values because each transaction approves three tips rather than two, clearing the tip pool more aggressively.

D. Security Results

Table III presents the probability p_2 of selecting an SPC tip under each algorithm, measured with $\lambda = 15$ and $\mu = 5$ (attacker controls 25% of network power). CATS and SURTS both reduce p_2 to below 0.001 across all tested SPC

TABLE III SPC TIP SELECTION
PROBABILITY p_2 (λ =15, μ =5)

SPC len.	URTS	MCMC$_1$	MCMC$_5$	S-URTS	G-IOTA	CATS
m =10	0.023	0.004	0.012	<0.001	0.053	<0.001
m =30	0.010	0.010	0.002	<0.001	0.056	<0.001
m =50	0.010	0.001	0.001	<0.001	0.053	<0.001

lengths, confirming that the absorbing Markov chain detection mechanism effectively identifies parasitic tips. G-IOTA exhibits the highest p_2 (0.053–0.056) across all chain lengths because its fairness mechanism treats SPC tips identically to honest tips: approving three tips per transaction clears the tip pool but provides no anomaly discrimination. URTS shows moderate vulnerability (p_2 = 0.010–0.023), while MCMC$_1$ and MCMC$_5$ reduce p_2 for longer chains ($m \geq 30$) as the biased random walk increasingly favors the higher-weight main DAG over the low-weight SPC.

E. Adaptive Response Dynamics

A key advantage of CATS over static algorithms is its ability to respond to the onset of an attack. In our experiments, when an SPC ($m = 30$) is attached at $t = 250$, CATS detects the anomalous tip within one PoW cycle via Module 2's Markov chain analysis, which assigns near-zero selection probability to the SPC tip. The composite signal $S(t)$ responds to the structural change, elevating $\alpha^{\sim}$ from $\alpha_{min} = 0.001$ to a peak of approximately 0.035 within 20 transactions. After the SPC tip is quarantined, $\alpha^{\sim}$ decays back toward α_{min} as the anomaly signals subside. Across 10 independent trials, CATS successfully quarantined the malicious tip in every run, with fewer than 0.1 honest tips falsely quarantined on average. The primary protection mechanism operates through the selection probability distribution (Table III): the absorbing
Markov chain assigns $D_i \approx 0$ to SPC tips regardless of detection timing, ensuring $p_2 < 0.001$ even when the SPC tip is briefly visible to honest transactions.

F. Parameter Sensitivity

The detection sensitivity parameter κ controls the tradeoff between false positives (honest tips quarantined) and false negatives (SPC tips undetected). Lower κ values produce a more aggressive threshold $T(t)$, increasing

detection coverage at the cost of quarantining honest tips. Higher κ values relax the threshold, reducing false positives but allowing short SPCs to evade detection. We tested $\kappa \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ and found that $\kappa = 1.5$ provides a stable balance: across our security experiments (Table III), CATS achieves $p_2 < 0.001$ while the adaptive response experiments show fewer than 0.1 honest tips falsely quarantined per run on average. The signal weights $\mathbf{w} = [0.3, 0.4, 0.3]$ shows lower sensitivity: the weight divergence signal σ_2 is assigned the highest weight because cumulative weight heterogeneity is the earliest observable indicator of parasitic substructure. A comprehensive quantitative sweep of κ and $\mathbf{w}$ with per-configuration error rates is deferred to future work.

G. Resilience to Adaptive Adversaries

CATS is designed to resist two classes of adaptive adversaries beyond the standard SPC. A *slow-build* adversary adds SPC transactions at a reduced rate (e.g., $\mu/3$) to avoid triggering sharp spikes in the weight divergence signal σ_2. By construction, the reduced attacker rate weakens the SPC's cumulative weight growth relative to the main DAG, which lowers the eventual selection probability of the SPC tip. Even if σ_2 responds slowly, the tip saturation signal σ_1 accumulates over time as the additional SPC tips increase the tip count ratio $\rho(t)$, eventually elevating $\tilde{\alpha}$. Moreover, Module 2's Markov chain analysis assigns near-zero selection probability to SPC tips regardless of their arrival rate, since the absorbing random walk has negligible probability of reaching a linearly chained tip through a single attachment point.

A *Sybil-augmented* SPC distributes the adversarial sub-DAG across multiple fake identities. Because Module 2 operates on the topological structure of the DAG rather than on identity labels, the chain-like branching pattern (branching factor ≤ 1.2) remains detectable regardless of how many identities generate it. A quantitative evaluation of detection rates and latencies under these adversarial strategies with the corrected simulator is left for future work.

V. Discussion

A. Complexity and Overhead

The per-transaction computational overhead is dominated by the absorbing Markov chain computation in Module 2 at $O(KN^2)$, where $K \leq 30$ iterations sufficed for convergence in all our experiments ($N = 500$ subtangle window). Under benign conditions, Module 2 can be executed every $\Delta t_r = 50$ transactions rather than per-transaction, reducing amortized cost to $O(|L(t)|)$. Space complexity is $O(N^2)$ for the transition matrix plus $O(|Q(t)|)$ for the quarantine set.

In practice, generating a 2000-transaction tangle over 100 runs at $\lambda = 15$ required approximately 5s for URTS, 7s for MCMC, 26s for S-URTS, and 28s for CATS on a consumer laptop (Apple M-series, Python 3.11). The overhead of CATS relative to S-URTS is modest ($< 10\%$), since both algorithms share the absorbing Markov chain computation; CATS adds only the signal calculation and quarantine management. These requirements are feasible for full nodes; lightweight IoT devices can receive pre-computed eligible tip sets from gateway nodes.

B. Applicability to IOTA 2.0

CATS is applicable to both IOTA 1.0-style systems (where TSA is the primary consensus mechanism) and IOTA 2.0 architectures (where TSA manages DAG topology while FPC handles conflict resolution) [13]. In the latter, CATS's anomaly detection complements the Mana reputation system [13]: Mana provides identity-level Sybil protection while CATS provides structure-level parasite chain protection. The variable reference count in IOTA 2.0 (two to eight approvals) can be integrated with Module 3 by allocating additional references to recovering tips.

C. Limitations

Formal convergence proofs for the feedback loop between Modules 1 and 2 are deferred to future work—the adaptive dynamics are validated empirically through simulation rather than proven analytically. The absence of a formal guarantee means that pathological parameter combinations or arrival patterns could, in principle, cause the $\tilde{\alpha}$ adaptation to oscillate or converge slowly, although no such behavior was observed in our experiments.

Network-layer effects (propagation latency, partial DAG views, eclipse attacks) are not modeled in our simulator; all nodes observe the same DAG state within each PoW window. In a real deployment, nodes with delayed or incomplete views may compute different tip distributions, potentially affecting detection consistency across the network.

The quantitative results for the parameter sensitivity sweep (Section IV-F) and the adaptive adversary experiments (Section IV-G) report qualitative design properties rather than exhaustive per-configuration error rates; a full quantitative evaluation of these scenarios is planned for future work. Additionally, the reference configuration is optimized for the parameter ranges tested ($\lambda \in [5, 20]$); deployment at significantly different scales may require recalibration of the signal weights $\mathbf{w}$, the detection sensitivity κ, and the normalization constants $\rho_{\max}$ and $\delta_{\max}$.

VI. Conclusion

This paper introduced CATS, an adaptive tip selection algorithm for DAG-based distributed ledgers that addresses the tension between security, scalability, and fairness. Through discrete-event simulation across multiple arrival rates and attack scenarios, we demonstrated that CATS achieves: tip count stability within 5% of URTS under benign conditions; SPC tip selection probability below 0.001 under attack; and low false positive rates with the recommended configuration (κ = 1.5, $\mathbf{w}$ = [0.3,0.4,0.3], γ = 0.15). The adaptive mechanism elevates $\tilde{\alpha}$ in response to parasitic substructure and restores scalability after the threat subsides, while the absorbing Markov chain analysis provides structure-level protection complementary to identity-based defenses. Future work will pursue formal convergence analysis of the adaptation dynamics, a comprehensive quantitative evaluation of parameter sensitivity and adaptive adversary resilience, integration with IOTA 2.0's Mana system, and evaluation under realistic network latency conditions.

References

[1] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*. IEEE, 2013, pp. 1–10.

[2] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: Challenges and solutions," *arXiv preprint arXiv:1608.05187*, 2016.

[3] S. Popov, "The tangle (version 1.3)," 2017, whitepaper. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

[4] Q. Wang, J. Yu, S. Chen, and Y. Xiang, "Sok: Diving into dag-based blockchain systems," *ACM Computing Surveys (CSUR)*, vol. 55, no. 12, pp. 1–37, 2022.

[5] F. Guo, A. Hecker, and S. Dustdar, "A scalable and secure transaction attachment algorithm for dag-based blockchain," *IEEE Internet of Things Journal*, vol. 12, no. 9, pp. 12298–12310, 2025.

[6] I. Amores-Sesar and C. Cachin, "We will dag you," *arXiv preprint arXiv:2311.03092*, 2023.

[7] S. Popov, O. Saa, and P. Finardi, "Equilibria in the tangle," *Computers & Industrial Engineering*, vol. 136, pp. 160–172, 2019.

[8] A. Cullen, P. Ferraro, C. King, and R. Shorten, "On the resilience of dagbased distributed ledgers in iot applications," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7112–7122, 2020.

[9] X. Lu, C. Jiang, and P. Wang, "A survey on consensus algorithms of blockchain based on dag," *IEEE Access*, vol. 11, pp. 3672–3689, 2024.

[10] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, 2015, pp. 129–144.

[11] C. Fan, "Performance analysis and design of an iot-friendly dagbased distributed ledger system," Master's thesis, University of Alberta, Edmonton, AB, Canada, 2019.

[12] G. Bu, Ö. Gürcan, and M. Potop-Butucaru, "G-iota: Fair and confidence aware tangle," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2019, pp. 166–170.

[13] N. Sealey, A. Aijaz, and B. Holden, "Iota tangle 2.0: Toward a scalable, decentralized, smart, and autonomous iot ecosystem," *IEEE Access*, vol. 10, pp. 98514–98535, 2022.