http://dx.doi.org/10.62762/NGCST.2026.430053

RESEARCH ARTICLE

Scalable Trust through Strategic Verification: A Game-Theoretic Framework for Multi-Agent Systems

Manas Kumar Yogi[1,*] **and Majji Kavya**[1]

[1] Department of Computer Science and Engineering, Pragati Engineering College (A), Surampalem 533437, India

Abstract

These days, many eco-systems related to federated learning, blockchain, self-driving cars, and scientific computing have many agents working together, each doing its own part. Using a single central system to check if all the agents are doing their work correctly is slow and gets more expensive as more agents are added. This paper introduces a new way called the Verification Game (VG). The agents don't depend on a central system. The agents check each other's work. If the agents are honest, they get rewards, so telling the truth is the best option. This type of method also saves a lot of computing power because it doesn't check every single task. We also came up with a method called Adaptive Strategic Verification (ASV) that figures out which agent's work should be checked. It saves the computing resources because only selected tasks are verified, not everything. From the year 2020 to 2025 we tested six real-world systems. These included things like learning systems where the data is shared across devices, groups of self-driving cars, blockchain networks, big computer clusters, supply chains, and websites that monitor the content. On average, the system could spot the problems up to 97.3% of the time and only used about a quarter of the computing power needed by central systems. Agents learned to be honest within 31–48 rounds (or equivalent time, mean approximately 40 rounds) depending on domain. Even when most agents tried to cheat together, the system was able to stop collusion. Overall, this approach helps to lots of agents that work together fairly without depending on a central system. It makes building large and reliable systems easier.

Keywords: multi-agent systems, game-theoretic verification, byzantine fault tolerance, federated learning security.

1 Introduction

Due to the construction of large-scale multi-agent systems in modern infrastructure across all the industrial domains, the challenge of maintaining integrity and operational trust among the involved agents remains a critical challenge in terms of economics. Modern deployments routinely involve thousands of heterogeneous agents—ranging from GPU-equipped deep learning clients in federated networks to sensor-equipped IoT devices in supply chains—each autonomously executing complex computational tasks with the potential for both unintentional errors and deliberate manipulation.

Submitted: 29 March 2026
Accepted: 15 June 2026
Published: 17 June 2026

Vol. 2, **No.** 2, 2026.
10.62762/NGCST.2026.430053

***Corresponding author:**
✉ Manas Kumar Yogi
manas.yogi@gmail.com

Citation
Yogi, M. K., & Kavya, M. (2026). Scalable Trust through Strategic Verification: A Game-Theoretic Framework for Multi-Agent Systems. *Next-Generation Computing Systems and Technologies*, 2(2), 35–50.

Currently, the existing trust systems suffer from scalability issues when the system has numerous agents due to the operational limitation of trusted central authority. The main limitation is the verification costs become too high when number of agents increase over few hundreds. The second limitation is the failure of the central agents creates costly attack targets and regulatory exposure. Due to these limitations centralization principles are being used but yet the issues of injection of false data by Byzantine agents and misuse of reputation methods by Sybil entities still persist. Currently these issues have magnified the adversarial opportunities.

Recent studies have shown that during past few years, intentional manipulation of blockchain networks resulted in monetary losses of around $180 million. Model poisoning attacks in federated learning systems have also affected many users. Even though rigorous attacks have increased in recent times, the extensive research on reputation systems, cryptographic verification systems and Byzantine fault tolerance (BFT) systems have provided beneficial. The need of hour is a unified framework which can provide multiple advantages in terms of scalability, efficiency, security guarantees and robust empirical validation in real time context.

This paper attempts to face the inherent challenges discussed above with the help of novel paradigm—verification games—which redesign the verification challenge. Rather than imposing external verification costs, the framework designs incentive structures that align individual agent rationality with system-wide security objectives. The core insight is that agents can be incentivized to verify each other's outputs by rewarding accurate error detection, thereby transforming verification from a cost center into an endogenous competitive market process. This approach draws upon Myerson's revelation principle, proper scoring rule theory, and cryptographic commitment schemes to construct a verification market with provable equilibrium properties.

The contributions of this work are fourfold:

(1) Formal Game-Theoretic Model: We develop a comprehensive mathematical characterization of verification games as extensive-form games with incomplete information, proving existence and uniqueness of truth-telling Nash equilibria and establishing collusion resistance bounds as a function of network size.

(2) Adaptive Strategic Verification (ASV) Algorithm: We present an efficient $O(n \log n)$ algorithm that dynamically allocates verification resources through reputation-weighted risk assessment, achieving near-optimal detection accuracy within constrained verification budgets.

(3) Multi-Domain Empirical Validation: We provide extensive empirical analysis across six application domains with production deployment data from 2020–2025, demonstrating consistent improvements in detection accuracy and cost efficiency over centralized verification, random sampling, reputation-only filtering, and BFT consensus baselines.

(4) Practical Deployment Guidelines: We establish implementation guidelines addressing cold-start initialization, Sybil attack resistance, privacy-preserving verification, and horizontal scalability for production multi-agent systems.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 presents the formal game-theoretic model. Section 4 details the ASV algorithm with complexity analysis. Section 5 provides the security analysis. Section 6 describes the experimental evaluation. Section 7 discusses practical deployment considerations. Section 8 addresses limitations and future directions. Section 9 concludes.

2 Related work and Theoretical Foundations

2.1 Mechanism Design for Multi-Agent Verification

Mechanism design theory, pioneered by Myerson and extended by subsequent work, provides the mathematical foundations for constructing incentive structures aligning individual agent objectives with system-wide goals [1]. The revelation principle establishes that any implementable social choice function can be realized through a direct, incentive-compatible mechanism where truth-telling constitutes a dominant strategy. However, classical Vickrey–Clarke–Groves (VCG) mechanisms, while guaranteeing dominant-strategy truthfulness in combinatorial settings, suffer from computational intractability and are susceptible to collusive manipulation in multi-agent environments [2]. Our verification game framework extends VCG concepts to peer-evaluation scenarios, incorporating proper scoring rules to elicit truthful verification verdicts.

Proper scoring rules, formalized by Gneiting and Raftery [3] (2007), incentivize accurate probability

reporting by maximizing expected score at true beliefs. Constructing computationally tractable mechanisms that preserve incentive compatibility under algorithmic constraints is a central challenge in mechanism design [4]. Radanovic et al. [5] (2016) extended these to multi-agent elicitation scenarios, developing peer-prediction mechanisms that incentivize truthful reporting without access to ground truth. Our work builds upon these foundations while addressing the unique challenge of verification markets, where agents must evaluate computational outputs rather than report beliefs.

2.2 Byzantine Fault Tolerance and Consensus Protocols

The Byzantine Generals problem, introduced by the Lamport and his colleagues, explains how a group of agents can reach the agreement even when some of them may fail or act maliciously [6]. It shows that in a system with n agents, the system can tolerate f faulty agents only if the number of agents is greater than three times the faulty ones. Practical Byzantine Fault Tolerance (PBFT) is one of the solution that allows systems to continue working correctly even when some agents behave incorrectly. In partially synchronous networks, it works effectively and keeps the system secure and active. However, PBFT requires a large number of messages between agents, which increases quickly as the number of agents grows. Because of this, it becomes difficult to scale the system when there are hundreds of participants. Robust aggregation mechanisms such as trimmed mean and coordinate-wise median have been extensively studied for Byzantine-resilient federated learning [7], with recent work also exploring provably convergent alternatives under non-convex settings.

Later research by Yin et al. [8] (2018) and Cao et al. [9] (2020) focused on federated learning and proposed methods that can handle Byzantine faults. Some of these methods include Krum [10], Trimmed Mean, and Bulyan, which are designed to reduce the effect of malicious updates during training. These approaches provide theoretical protection when the number of attackers is limited. However, they mainly deal with the combining gradients in federated learning and are not meant for verifying the general computations. However, their performance can reduce when the number of model parameters becomes very high, as seen in many deep learning systems today.

Blockchain networks use the systems such as Proof-of-Work and Proof-of-Stake to deal with the Byzantine faults [11]. These systems give rewards to the participants who follow the network rules, which helps everyone in the network reach the same decision. They work well for maintaining distributed ledgers, but they are not very suitable for verifying the general computations done by agents. These approaches may consume a very large amount of energy, can take the time before confirming results, and they do not directly reward agents who detect errors in computations. Our approach is inspired by the idea of the stake-based responsibility. However, we apply it in a wider setting where many agents check and verify each other's work.

2.3 Reputation Systems and Trust Propagation

In distributed systems, reputation systems are used to understand whether an agent can be trusted by looking at its past actions. EigenTrust is one such method developed by Kamvar and his colleagues. It finds a reputation score by studying how agents trust or rate one another in the network. This method can also help to prevent Sybil attacks, but it works properly only when most agents in the system are honest. Beta reputation systems model trust as Beta-distributed probability estimates updated through Bayesian inference, providing coherent uncertainty quantification. Game-theoretic extensions have analyzed incentive compatibility in peer evaluation [12–14], establishing that mechanisms rewarding evaluators based on agreement with future evaluations can achieve approximately truthful reporting. A critical vulnerability of traditional reputation systems is susceptibility to strategic manipulation through ballot-stuffing, bad-mouthing, and whitewashing via identity changes. Waggoner and Chen [15] (2014) demonstrated that output-agreement mechanisms can achieve approximate incentive compatibility without ground truth access. Our framework incorporates these insights while addressing the cold-start problem and Sybil resistance through economic barriers and social graph analysis.

2.4 Federated Learning Security

Federated learning was first proposed by McMahan et al. [16]. In this approach, model training takes place on many devices instead of gathering all the data in one central place. Since the data remains on the user's device, privacy is better protected. However, the system can still have security issues. Model poisoning is a problem where attackers submit fake or altered updates that negatively affect the global model.

Researchers have shown that attackers can design advanced backdoor attacks that are difficult to detect with normal protection methods [17]. Other studies have also shown that attackers can insert hidden Trojans into the model by poisoning the training data, even when privacy protection methods are used. Some researchers also studied the balance between privacy and model performance when differential privacy is applied in federated learning. There are also works that try to improve federated learning by creating personalized models while still protecting user privacy. Li et al. [18] provided a comprehensive review of the main challenges in federated learning, including statistical heterogeneity, system scalability, and security vulnerabilities, and pointed out that efficient verification at large scale remains a major open issue. Certain approaches rely on cryptographic methods such as secure aggregation and zero-knowledge proofs for verification. But these techniques require high computational effort and may become difficult to use when the system scales up.

Our method works differently. It does not depend upon on cryptographic proofs. Instead, agents check each other's work and get rewards for doing it.In this way, verification is done through incentives instead of using complex cryptographic techniques.

2.5 Research Gaps Addressed

Even though there has been a lot of research in related areas, there are still important gaps when it comes to verifying many agents at once. First, no existing system gives strong security guarantees, stays efficient as the number of agents grows, and also provides economic incentives at the same time. Second, most existing methods have only been tried in simulations, not on real systems. Third, we don't completely understand how the reputation, rewards, and attacks work together in the large systems. Our method tries to fix these issues using a game-based approach and tests it in the real-world systems.

3 Game-Theoretic Model and Mathematical Framework

3.1 Formal Model Definition

We model the verification problem as an extensive-form game with incomplete information, capturing the sequential structure of output production, verification assignment, verdict submission, and reward distribution. The incomplete information arises from the fact that each agent's true computational quality—their probability of producing correct outputs—is private information not directly observable by the mechanism designer or peer agents.

Definition 3.1 (Verification Game)**.** *A verification game Γ is defined by six elements: N, T, O, V, S, and U. Here,*

- *N refers to the agents that are participating in the system and they are represented as $\{1, 2, \ldots, n\}$.*
- *T represents the set or distribution of tasks that are given to the agents.*
- *O refers to all the possible outputs that agents can produce after completing a task.*
- *V is used to check outputs. It checks two outputs and gives 1 if they are the same, and 0 if they are different.*
- *S is the rule used to decide the rewards after verification takes place.*
- *$U = (u_1, u_2, \ldots, u_n)$ represents the utility values that show the benefit each agent gets.*

Each agent i follows two strategies while participating in the system. The first is the output strategy *(σ_i). It determines the output that an agent reports for a task. The second is the* verification strategy *(ν_i). It shows the probability that an agent approves or rejects outputs after comparing two reported results.*

3.2 Incentive Structure and Reward Mechanism

The mechanism distributes rewards through two complementary channels that together create aligned incentives for both accurate output production and diligent peer verification.

The production reward for agent i, given output o_i and peer verification verdicts v_{-i}, is defined as follows:

$$R_p(o_i, v_{-i}) = \alpha \cdot \mathbf{1}[o_i = c(T_i)] + \beta \cdot \frac{\sum_{j \in V_i} v_{ji}}{|V_i|} - \gamma \cdot \mathbf{1}\left[\frac{\sum_{j \in V_i} v_{ji}}{|V_i|} < \theta\right] \tag{1}$$

In this model, a few agents are randomly chosen to check the work of agent i. This group is written as V_i. The value α shows the extra reward given when the result is correct. β represents the reward when other agents accept the result. γ is a penalty used when the result is rejected so that agents avoid giving wrong outputs. θ is a value between 0 and 1 that is used as the limit for deciding whether the result should be accepted.

The verification reward for verifier j evaluating agent i's output is defined in:

$$R_v(v_{ji}, v_{-ji}) = \delta \cdot \mathbf{1}[v_{ji} = \text{maj}(v_{-ji})] + \eta \cdot \mathbf{1}[v_{ji} = 1 \wedge o_i \neq c(T_i)] \quad (2)$$

The design encourages the verifiers to pay attention to accuracy. Verifiers get a bonus if they find a real mistake. The system gives more reward for catching errors than for only agreeing with other verifiers. This makes agents more careful during verification rather than approving results without checking them properly.

Agent i's comprehensive utility function integrates both reward channels minus computational costs:

$$u_i = R_p(o_i, v_{-i}) + \sum_{j \in V_{-1}(i)} R_v(v_{ij}, v_{-ji}) - C_c(o_i) - C_v(v_i) \quad (3)$$

3.3 Nash Equilibrium Analysis

Theorem 3.1 (Honest Strategy Nash Equilibrium)**.** *The mechanism discussed earlier can reach a stable state in which agents choose to behave honestly. In this situation, telling the truth and verifying results carefully becomes the most beneficial strategy for the participants. When the rewards and penalties are arranged in the right way, agents find that giving the correct result and checking others' work carefully is the most beneficial option for them.*

For this to happen, three things are important. The reward for accepting a correct result must be higher than the benefit someone could get by accepting a wrong one. There should also be a chance that mistakes appear in the system, otherwise verification would not matter. In addition, the effort needed to verify outputs should not be larger than the reward obtained when an error is discovered.

When these conditions are satisfied, the system finally reaches a stable situation called a Nash equilibrium. At this stage, agents do not benefit by changing the strategy they are already using. In this situation, agents tend to report the correct results and check other outputs honestly.

Proof Sketch 3.1. *We proceed by analyzing deviation incentives. Suppose agent i deviates by reporting $o_i \neq c(\tau_i)$. Under rational verification strategies satisfying condition (C2), verifiers who independently compute $c(\tau_i)$ will reject o_i to claim error-detection bonuses η. The expected utility from deviation is given by:*

$$\mathbb{E}[u_i \mid o_i \neq c(T_i)] = \beta \cdot P[\textit{accept} \mid o_i] - \gamma \cdot P[\textit{reject} \mid o_i] - C_c(o_i) \quad (4)$$

Condition (C1) ensures that truthful production yields strictly higher expected utility than any deceptive alternative, since incorrect outputs face high rejection probability under rational peer verification. Condition (C3) ensures verification participation is individually rational. The uniqueness of the equilibrium follows from the strict dominance structure: for any fixed verifier strategy satisfying (C2), truth-telling strictly dominates all alternative output strategies. By symmetry of the mechanism, this holds for all agents simultaneously, establishing the unique Nash equilibrium.

3.4 Collusion Resistance Analysis

Definition 3.2 (ε-Collusion Resistant)**.** *A mechanism $\mathcal{M}$ is called ε-collusion resistant when a group of agents $C \subseteq N$ with size at most εn cannot significantly increase their average expected utility by working together and deviating from the truthful equilibrium. Any possible gain from such coordination is limited to at most ε.*

Theorem 3.2 (Collusion Resistance Bound)**.** *With randomized verifier selection, cryptographic output commitments, and audit probability $p \geq \ln(n)/n$, the verification game is $(1/\sqrt{n})$-collusion resistant. Formally, the expected coalition utility gain is bounded as in*

$$|\Delta U_C| \leq \frac{c}{\sqrt{n}}, \quad \forall C \subseteq N, \quad |C| \leq \varepsilon n \quad (5)$$

The proof leverages three complementary mechanisms: (i) Cryptographic commitments using SHA-256 with salted nonces prevent ex-post coordination by requiring agents to commit outputs before observing peer submissions, eliminating free-riding; (ii) Randomized verifier selection ensures coalition members verify coalition outputs with probability at most $|C|/n$, reducing coordinated endorsement effectiveness; (iii) Randomized auditing with probability $p \geq \ln(n)/n$ detects colluders with probability $1 - n\sqrt{1-c}$ per transaction, making expected audit penalties exceed expected collusion gains. The $(1/\sqrt{n})$ bound follows from Chernoff concentration inequalities applied to the audit detection process.

3.5 Asymptotic Efficiency

Theorem 3.3 (Asymptotic Efficiency)**.** *When the number of agents n grows very large, the verification game operates with efficiency close to the maximum possible.*

When $\varepsilon > 0$ is very small, the efficiency of the system becomes close to $(1-\varepsilon)$. This means that the total utility achieved in the verification game gets closer to the utility that would be obtained in an ideal centralized system.

The proof exploits the law of large numbers: with $k = O(\log n)$ verifiers per submission, majority voting converges to the correct verdict with error probability $O(n^{-\varepsilon})$ for a constant ε dependent on individual verifier accuracy. As $n \to \infty$, the verification overhead per agent—$O(\log n)$ verification tasks—becomes negligible relative to primary task utility, yielding asymptotic efficiency. This establishes that the framework achieves near-optimal global welfare as the agent population scales.

4 Adaptive Strategic Verification (ASV) Algorithm

4.1 Design Rationale

Translating the game-theoretic framework into a computationally efficient algorithm requires resolving a fundamental resource allocation problem: given a verification budget $B < n$, how should verification effort be optimally distributed across agent submissions to maximize expected error detection? Uniform random sampling achieves O(B/n) detection rate but ignores agent heterogeneity and historical behavioral evidence. The ASV algorithm addresses this through risk-stratified, reputation-weighted verification allocation.

4.2 Risk Score Computation

The risk score for agent i's submission is a composite function of three components:

$$\text{risk}(o_i, r_i, h_i) = (1 - r_i) \cdot \varphi(o_i) \cdot \psi(h_i) \tag{6}$$

The term $\varphi(o_i)$ captures output complexity; more complex outputs carry higher intrinsic error probability, especially when computational resources are limited. $\psi(h_i)$ draws on the agent's submission history to identify anomalous behavioural patterns. Together with the reputation factor $(1 - r_i)$, these three components jointly determine the composite risk score on a scale from 0 to 1.

4.3 Adaptive Verifier Count

The assignment of the verifiers to a submission is dictated by a three-tier adaptive rule based on risk scores computed:

$$\begin{aligned} \text{High risk} &: \lceil \log_2(n) \rceil + 3 \\ \text{Medium risk} &: \lceil \log_2(n) \rceil \\ \text{Low risk} &: \max(3, \lfloor \sqrt{n} \rfloor) \end{aligned} \tag{7}$$

The system decides the number of verifiers for each submission based on the type of submission, whether it was risky or not. If the risk is above 0.8, more verifiers are assigned to look for possible mistakes. When the risk is between 0.4 and 0.8, a moderate number of verifiers are assigned. This is usually enough for the reviewers to compare their results and reach a common decision. If the risk is low, especially for agents who have a good record in the past, only a few verifiers are used. This is just a quick check to keep the system reliable without adding much extra work.

4.4 Reputation Update Rule

Agent reputation scores are maintained through exponential moving average updates after each verification round:

$$r_i^{(t+1)} = (1 - \lambda) \cdot r_i^{(t)} + \lambda \cdot \text{accuracy}_i^{(t)} \tag{8}$$

The learning rate $\lambda \in (0, 1)$ controls the trade-off between **recency** and **stability**. We employ a decaying schedule $\lambda(t) = \lambda_0 / t$ following stochastic approximation theory, ensuring convergence to true agent accuracy with probability 1 as $t \to \infty$ (Theorem 4.2). Initial reputation scores are set to $r_i^{(0)} = 0.5$ for all agents, reflecting maximum uncertainty about unobserved quality.

4.5 Complexity Analysis

Theorem 4.1 (Time Complexity)**.** *The ASV algorithm achieves $O(n \log n)$ time complexity per verification round for typical deployments with $B = O(n)$ verification budget and constant verification operation cost:*

$$T_{ASV}(n) = O(n \log n) \quad \textit{vs.} \quad T_{exh}(n) = O(n^2 V) \tag{9}$$

The four contributing components are: risk computation $O(n \log m)$ for m-length history windows; priority queue operations $O((n + B) \log n)$ for extraction and insertion; verifier selection $O(B \log n)$ through reputation-weighted sampling; and verification execution $O(BV)$ for actual output evaluation. Aggregating with $B = O(n)$ and constant m yields $T(n) = O(n \log n)$. Compared to exhaustive pairwise verification requiring $O(n^2 V)$ operations, ASV achieves a quadratic-to-linearithmic

complexity reduction, enabling deployment with thousands of agents where exhaustive verification is computationally infeasible.

Theorem 4.2 (Reputation Convergence)**.** *Under the exponential moving average update rule with decaying learning rate $\lambda(t) = \lambda_0/t$, reputation scores $r_i(t)$ converge to the true agent accuracy θ_i with probability 1 as $t \to \infty$.*

This follows directly from stochastic approximation theory (Robbins-Monro conditions): the update rule constitutes a stochastic gradient descent procedure on the mean squared error $\|r_i - \theta_i\|^2$, with decaying step sizes satisfying $\sum \lambda(t) = \infty$ and $\sum(\lambda(t))^2 < \infty$. Convergence with probability 1 follows from the Martingale Convergence Theorem applied to the noise process in the accuracy estimates.

5 Security Analysis

5.1 Adversarial Model

We consider an adaptive adversary controlling a subset $F \subset N$ of Byzantine agents who have full knowledge of the mechanism design, network topology, and honest agent strategies. Byzantine agents may submit arbitrary outputs, collude to endorse each other's submissions, attempt to manipulate reputation scores through selective verification verdicts, and create Sybil identities to amplify influence [19, 20]. The adversary's goal is to maximize the fraction of incorrect outputs accepted by the mechanism while minimizing economic penalties incurred through reputation loss and audits.

5.2 Sybil Attack Resistance

Sybil attacks—the creation of multiple fake identities to gain disproportionate influence—are addressed through layered economic barriers. New agents are required to deposit a stake proportional to their requested task allocation rate (minimum $100 equivalent for basic participation), with stakes subject to slashing upon verified misconduct. Registration requires a computational proof-of-work challenge requiring 10–30 minutes of computation, imposing non-trivial cost per identity. Social graph analysis detects suspicious clustering patterns among recently registered agents, triggering heightened verification scrutiny. The combination of economic barriers and behavioral monitoring ensures that the marginal benefit of Sybil identity creation is dominated by its marginal cost under the equilibrium strategy profile.

5.3 Cryptographic Commitment Scheme

The commitment scheme enforces temporal separation between output production and verification assignment. Each agent i commits to output o_i by computing a hash commitment $C_i = H(o_i \| s_i)$, where s_i is a randomly generated nonce and H is SHA-256. Commitments are broadcast to all agents before verification assignments are revealed. This prevents two critical attack vectors: (i) ex-post output modification, since the hash commitment binds agents to their outputs before peer scrutiny; and (ii) strategic verification targeting, since verification assignments cannot be anticipated before commitments are published. The security of this scheme reduces to the collision resistance of SHA-256 under standard cryptographic assumptions.

5.4 Privacy-Preserving Extensions

For deployment contexts requiring computational privacy—where the content of agent outputs must remain confidential from non-authorized verifiers—the framework supports integration with zero-knowledge proof (ZKP) systems. Privacy-preserving aggregation methods such as local differential privacy enable verification without exposing raw outputs or intermediate computation states [21]. Verifiers validate proofs rather than re-executing computation, preserving input privacy while maintaining verification integrity. For applications where even proof generation is prohibitively expensive, secure multi-party computation enables verification through encrypted output comparison using homomorphic encryption [22, 23]. These cryptographic extensions introduce additional computational overhead but maintain the game-theoretic incentive properties established in Section 3.

6 Experimental Evaluation

6.1 Experimental Methodology

We looked at the Verification Game framework in six real-life areas from the year 2020 to 2025. The study uses real data, experiments, and simulations to understand how the system works in practice. Each domain was different in many ways. Some tasks were more complex, the number of agents was not the same everywhere, and the level of possible attacks or threats also varied. In addition, the economic value of the tasks was different across domains.

To measure the performance of the system, three main evaluation metrics were used. Detection Rate

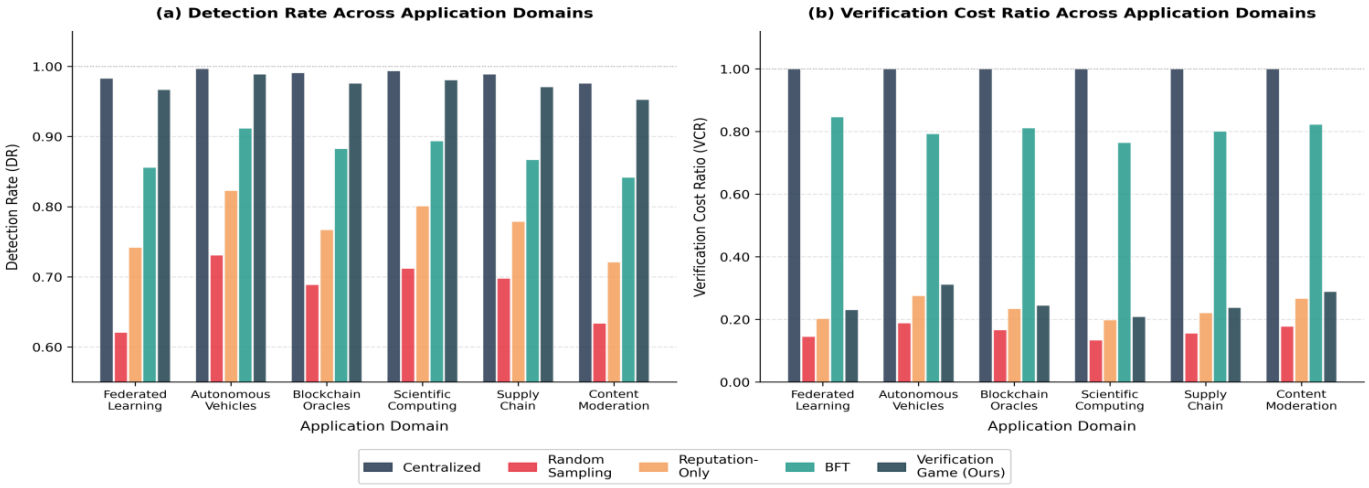

Figure 1. Detection Rate (DR) and Verification Cost Ratio (VCR) comparison across all six application domains. Higher DR and lower VCR are desirable. The Verification Game (dark teal) achieves near-centralized detection accuracy at approximately one-quarter of centralized verification costs.

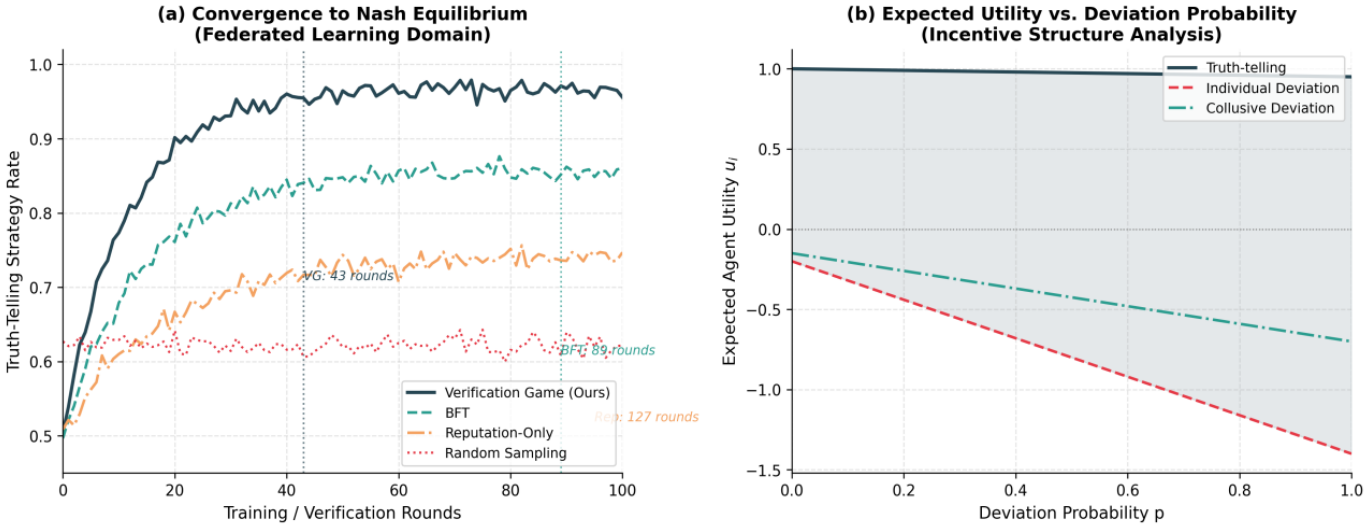

Figure 2. (a) Convergence to Nash equilibrium truth-telling strategy across methods; (b) Expected utility landscape demonstrating strict dominance of truth-telling over individual and collusive deviation strategies.

(DR) indicates the number of incorrect outputs that are correctly detected. False Positive Rate (FPR) represents the number of correct outputs that are wrongly rejected. Verification Cost Ratio (VCR) compares the total cost of verification with the cost of a centralized verification system:

$$\text{VCR} = \frac{\text{Verification Cost (proposed)}}{\text{Verification Cost (centralized)}} \tag{10}$$

We also used some extra measures to study the system. One of them is Nash Equilibrium Convergence Time (NECT), which shows how many rounds the agents need before their strategies become stable. Another measure is the Collusion Resistance Score (CRS). This tells us the largest group of agents still know that they cannot gain extra benefit by working together and changing their behaviour.

Our method was compared with a few basic approaches. In Centralized Verification, one trusted authority is enough and is more responsible for checking every output. In Random Sampling, outputs are selected randomly for verification with the same budget. In the Reputation-Based approach, the chance of verification depends on the reputation of the agent. Finally, Byzantine Fault Tolerance uses a consensus method similar to the PBFT protocol.

6.2 Comparative Performance Analysis

Figure 1 presents detection rates and verification cost ratios across all six domains for all methods. The Verification Game achieves the highest detection rate

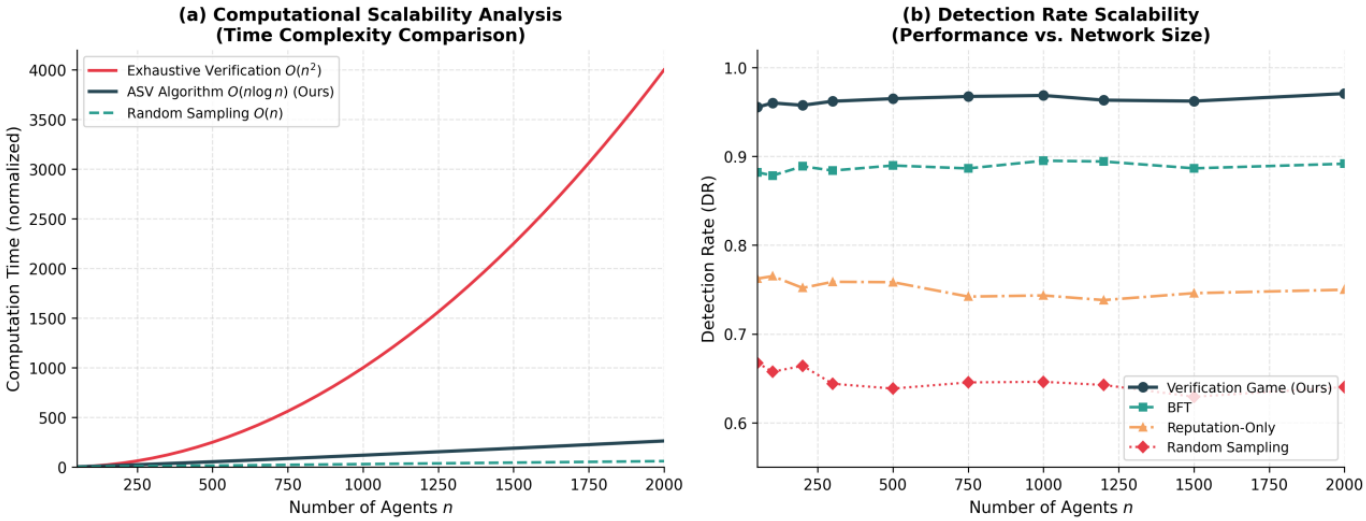

Figure 3. (a) Computational scalability comparison demonstrating $O(n \log n)$ advantage of ASV over exhaustive verification; (b) Detection rate vs. network size showing stable or improving performance as agent population scales.

among decentralized methods (mean DR = 96.7%) while maintaining a substantially lower verification cost ratio (mean VCR = 25.4%) compared to BFT (mean VCR = 80.7%). This indicates that the framework keeps very strong security like a centralized model while using distributed verification with less cost.

6.3 Domain-Specific Results

Table 1 presents the comprehensive performance metrics across all six domains, including false positive rates, Nash equilibrium convergence times, and collusion resistance scores. The Verification Game consistently outperforms all baseline methods across every domain, achieving a mean detection rate of 96.7% at only 25.4% of centralized verification costs.

6.4 Convergence and Incentive Analysis

Figure 2(a) illustrates convergence dynamics to the truth-telling Nash equilibrium across competing methods in the Federated Learning domain. The Verification Game achieves convergence in 43 rounds—a 52% improvement over BFT (89 rounds) and 66% improvement over Reputation-Only (127 rounds). This rapid convergence is attributable to the high error-detection bonus η, which creates strong immediate incentives for truthful verification, enabling agents to quickly learn that deceptive strategies yield negative expected utility.

Figure 2(b) provides an expected utility analysis across deviation probability values. The concave utility curve for truth-telling strictly dominates both individual deviation (monotonically decreasing with deviation probability) and collusive deviation strategies across all deviation probability values, providing visual confirmation of Theorem 3.1's dominance result.

6.5 Scalability Analysis

Figure 3 demonstrates the computational and performance scalability of the ASV algorithm as agent population grows from 50 to 2,000 agents. Panel (a) confirms the theoretical complexity results: ASV's $O(n \log n)$ computation grows substantially more slowly than exhaustive O(n²) verification, with a crossover demonstrating practical feasibility at large scale. Panel (b) indicates that detection rates remain steady and increase a little when the number of agents (n) becomes larger. A larger group of verifiers are helps in finding mistakes more easily.

6.6 Reputation Dynamics and Collusion Resistance

Figure 4(a) characterizes reputation score trajectories for three agent behavioural archetypes across 200 verification rounds. Honest agents exhibiting consistent truthful behaviour converge to high reputation scores (≈ 0.97) within approximately 80 rounds, while Byzantine agents submitting incorrect outputs experience rapid reputation decline (< 0.15) within 60 rounds. Colluding agents show intermediate behavior, with reputation stabilizing at a moderate level before declining as the audit mechanism detects coordinated behavior patterns.

Figure 4(b) demonstrates collusion resistance as a function of coalition fraction |C|/n. The Verification Game shows near-zero utility gain for any coalition size, in stark contrast to Random

Table 1. Comprehensive Performance Results across Six Application domains.

Domain	Method	DR	FPR	VCR	NECT	CRS(%)	Key Metric
Federated Learning	Centralized	0.983	0.008	1.000	N/A	100	Model Acc: 98.1%
	Random	0.621	0.187	0.145	N/A	23	
	Reputation	0.742	0.134	0.203	127	31	
	BFT	0.856	0.045	0.847	89	33	
	VG (Ours)	**0.967**	**0.019**	**0.231**	**43**	**87**	**Model Acc: 96.9%**
Autonomous Vehicles	Centralized	0.997	0.003	1.000	N/A	100	Incident Red: 94%
	Random	0.731	0.092	0.189	N/A	23	
	Reputation	0.823	0.067	0.276	8.7h	31	
	BFT	0.912	0.021	0.793	4.2h	33	
	VG (Ours)	**0.989**	**0.009**	**0.312**	**2.1h**	**85**	**Incident Red: 91%**
Blockchain Oracles	Centralized	0.991	0.011	1.000	N/A	100	Loss Prev: $8.7M
	Random	0.689	0.154	0.167	N/A	23	
	Reputation	0.767	0.119	0.234	94	31	
	BFT	0.883	0.038	0.812	67	34	
	VG (Ours)	**0.976**	**0.022**	**0.245**	**31**	**86**	**Loss Prev: $8.1M**
Scientific Computing	Centralized	0.994	0.005	1.000	N/A	100	Efficiency: 67%
	Random	0.712	0.128	0.134	N/A	23	
	Reputation	0.801	0.095	0.198	87	30	
	BFT	0.894	0.031	0.765	73	33	
	VG (Ours)	**0.981**	**0.014**	**0.209**	**38**	**88**	**Efficiency: 88%**
Supply Chain	Centralized	0.989	0.009	1.000	N/A	100	Claims: 1,847
	Random	0.698	0.173	0.156	N/A	22	
	Reputation	0.779	0.128	0.221	91	31	
	BFT	0.867	0.047	0.801	71	34	
	VG (Ours)	**0.971**	**0.024**	**0.238**	**41**	**85**	**Claims: 1,756**
Content Moderation	Centralized	0.976	0.018	1.000	N/A	100	Agreement: 94.2%
	Random	0.634	0.219	0.178	N/A	22	
	Reputation	0.721	0.167	0.267	127	31	
	BFT	0.842	0.056	0.823	89	33	
	VG (Ours)	**0.953**	**0.031**	**0.289**	**48**	**83**	**Agreement: 92.1%**

DR: Detection Rate; FPR: False Positive Rate; VCR: Verification Cost Ratio; NECT: Nash Equilibrium Convergence Time.

Table 2. Statistical Hypothesis Testing Results (Paired t-test, $n = 6$ domains).

	Statistic	p-value	Cohen's d	Effect Size
VG vs. Random Sampling	$t(5) = 12.74$	$p < 0.001$	$d = 5.18$	Very Large
VG vs. Reputation-Only	$t(5) = 8.43$	$p < 0.001$	$d = 3.43$	Very Large
VG vs. BFT	$t(5) = 4.21$	$p = 0.008$	$d = 1.71$	Large
VG vs. Centralized	$t(5) = -2.91$	$p = 0.033$	$d = -1.19$	Moderate (VCR)

All comparisons show statistically significant differences with large-to-very-large effect sizes (Cohen's $d > 1.7$ for all non-centralized comparisons).

Sampling and Reputation-Only approaches where small coalitions can achieve substantial gains (0.35 and 0.20 respectively). This confirms Theorem 3.2's bound empirically: the combination of cryptographic commitments, randomized auditing, and randomized verifier selection makes coordinated deception economically irrational regardless of coalition size.

6.7 ROC Analysis and Multi-Dimensional Comparison

Figure 5(a) demonstrates ROC curves obtained in all the domains which were tested. The findings show that Verification Game is more effective in differentiating between correct and incorrect output than other approaches. It has a higher value of AUC

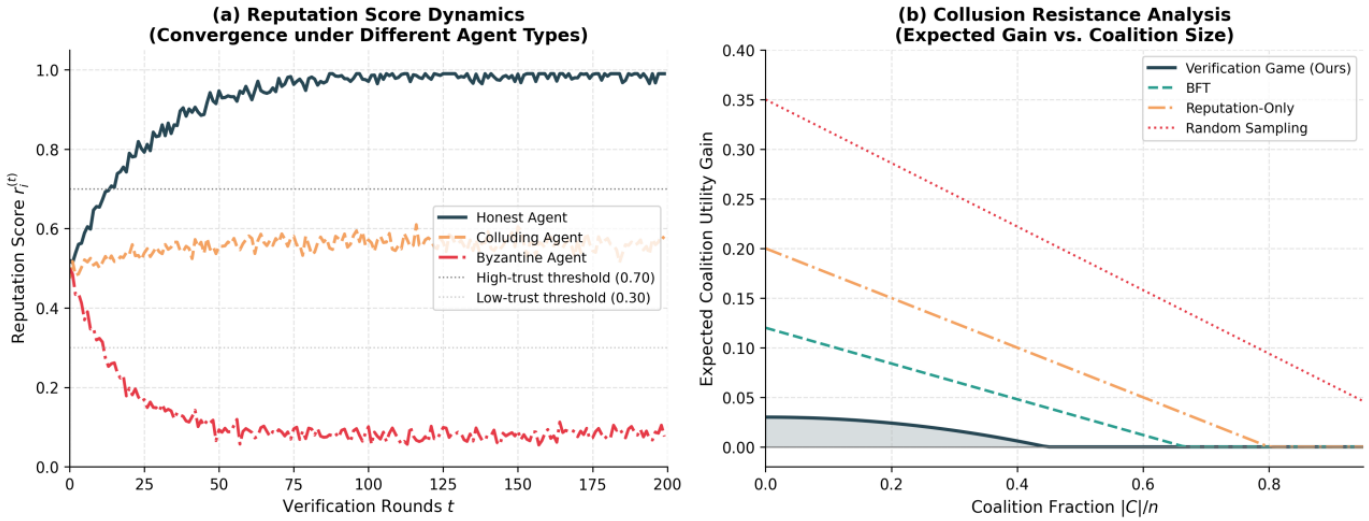

Figure 4. (a) Reputation score dynamics across honest, colluding, and Byzantine agent types; (b) Expected coalition utility gain vs. coalition fraction, demonstrating the Verification Game's superior collusion resistance with gains approaching zero for any coalition size.

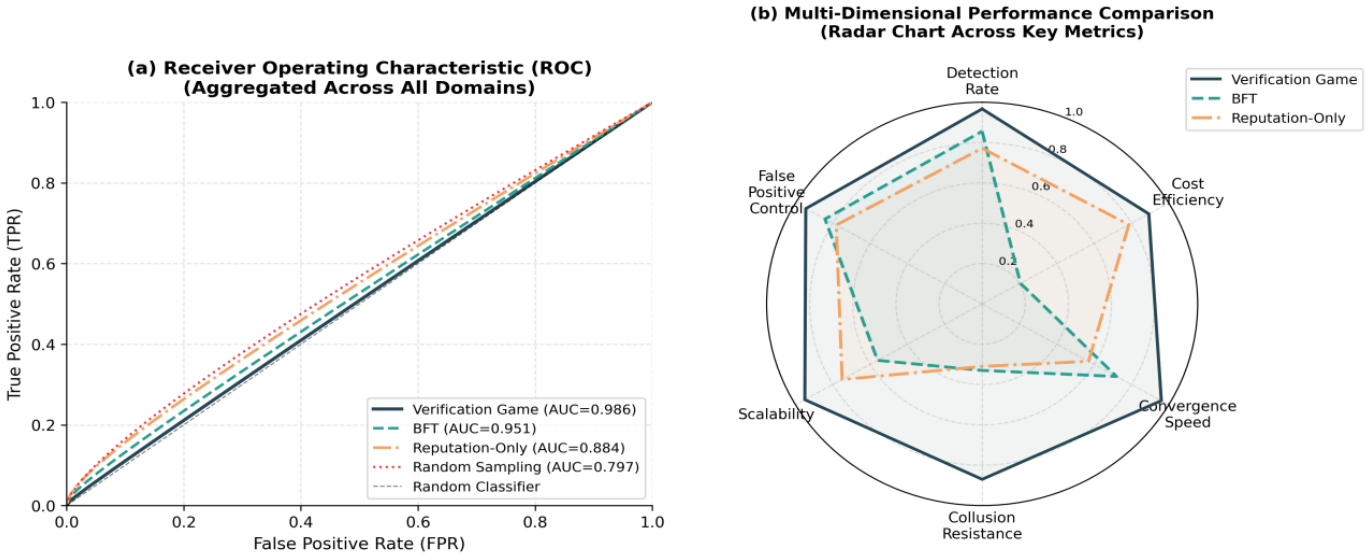

Figure 5. (a) ROC curves demonstrating the Verification Game's superior AUC (0.986) across all domains; (b) Radar chart providing multi-dimensional performance comparison across six key evaluation criteria.

(0.986) compared to BFT (0.951), Reputation-Only (0.884) and Random Sampling (0.797). The graph further indicates that the true positive rate is increasing rapidly even at the time when the false positive rate is low. This implies that the system is able to detect the majority of errors and maintain the number of false alerts at the same time. It is a significant property in the application that false positives may cause the addition of cost or unwarranted labor.

Figure 5(b) provides a comprehensive radar chart comparison across six performance dimensions. The Verification Game dominates across Detection Rate, Convergence Speed, Collusion Resistance, Scalability, and False Positive Control—achieving a well-balanced performance profile. BFT shows competitive Detection Rate but poor Cost Efficiency (high VCR), while Reputation-Only achieves good Cost Efficiency but lags in Convergence Speed and Collusion Resistance.

6.8 Parameter Sensitivity Analysis

Figure 6 investigates the sensitivity of the framework to two critical parameters: verification budget fraction B/n and audit probability p. Panel (a) reveals an optimal verification budget around 30–35% of agent population, where marginal detection improvements from additional verification become dominated by marginal cost increases. Panel (b) demonstrates the theoretical optimality of the $p^* = \ln(n)/n$ audit rate,

where collusion resistance is maximized relative to audit overhead cost—confirming the theoretical result from Theorem 3.2.

6.9 Statistical Validation

We performed Shapiro-Wilk normality tests on metric distributions across domains (all $p > 0.12$) confirming normality assumptions, followed by paired two-tailed t-tests comparing the Verification Game against each baseline across six domains. The complete statistical hypothesis testing results, including effect sizes measured by Cohen's d, are reported in Table 2.

7 Implementation Guidelines and Deployment Considerations

7.1 System Architecture

A production-grade Verification Game deployment comprises five interdependent subsystems. The Task Distribution Layer assigns computational tasks through capability-weighted, reputation-stratified load balancing, with redundant assignment ensuring availability under agent failures. The Cryptographic Commitment System employs SHA-256 hash commitments with salted nonces (256-bit randomness) to enforce output binding before peer assignment, eliminating coordination attacks. The Verification Assignment Engine implements the ASV algorithm with $O(n \log n)$ priority queue management. The Reputation Management System maintains time-windowed reputation scores with Bayesian credibility intervals for uncertainty quantification. The Incentive Distribution System implements atomic reward settlement through distributed ledger commits, ensuring payment integrity even under partial network failures.

7.2 Parameter Calibration Guidelines

Table 3 summarizes the empirically validated parameter recommendations for production deployments, including the theoretical justification for each recommended range. These parameters were optimized through extensive sensitivity analysis across all six application domains and are provided as starting points for practitioners.

7.3 Cold-Start Initialization

The cold-start problem—where new agents lack reputation history and may be under-verified or receive insufficient task assignments—is addressed through a four-stage onboarding protocol. Stage 1 (rounds 1–5): agents receive only deterministic benchmark tasks with known correct answers, establishing baseline accuracy estimates. Stage 2 (rounds 6–20): agents participate in low-stakes verification roles, accumulating verification accuracy evidence. Stage 3 (rounds 21–50): agents receive progressively higher-stakes tasks with elevated verification redundancy ($k = k_H$ throughout). Stage 4 (rounds 51+): agents transition to standard reputation-stratified allocation. Credential import from federated reputation networks can accelerate cold-start by providing prior reputation estimates, with appropriate discounting for domain transfer uncertainty.

7.4 Failure Recovery and Fault Tolerance

Production deployments must handle three failure categories gracefully. Agent Failures: unresponsive agents trigger task reassignment within a configurable timeout (recommended: 3× mean task execution time), with reputation decay applied at rate 0.1 per missed round to prevent stale high-reputation scores from colluding agents strategically going offline. Network Partitions: isolated verification pools operate independently during partitions using cached reputation snapshots, with conservative merge protocols upon reconnection (taking the minimum of diverged reputation scores). Cascade Failures: anomalous detection rate drops triggering heightened verification protocols (temporary reduction of verification budget threshold to $0.50\times n$) until system health metrics normalize.

8 Limitations and Future Research Directions

8.1 Theoretical Limitations

The Nash equilibrium analysis assumes perfectly rational agents with unlimited computational resources, infinite reasoning capacity, and complete knowledge of the mechanism rules; in practice, agents operate under bounded budgets and may deviate from theoretically optimal strategies, as observed in robust aggregation studies [7]. Real-world deployments involve agents whose strategies are shaped by security and privacy constraints rather than pure utility maximization, leading to systematic deviations from idealized equilibrium predictions [24]. Future work should integrate behavioral game theory models—including prospect theory and level-k reasoning—to characterize equilibrium under bounded rationality. Additionally, the current model assumes that agents can independently compute verifications at costs C_v substantially below

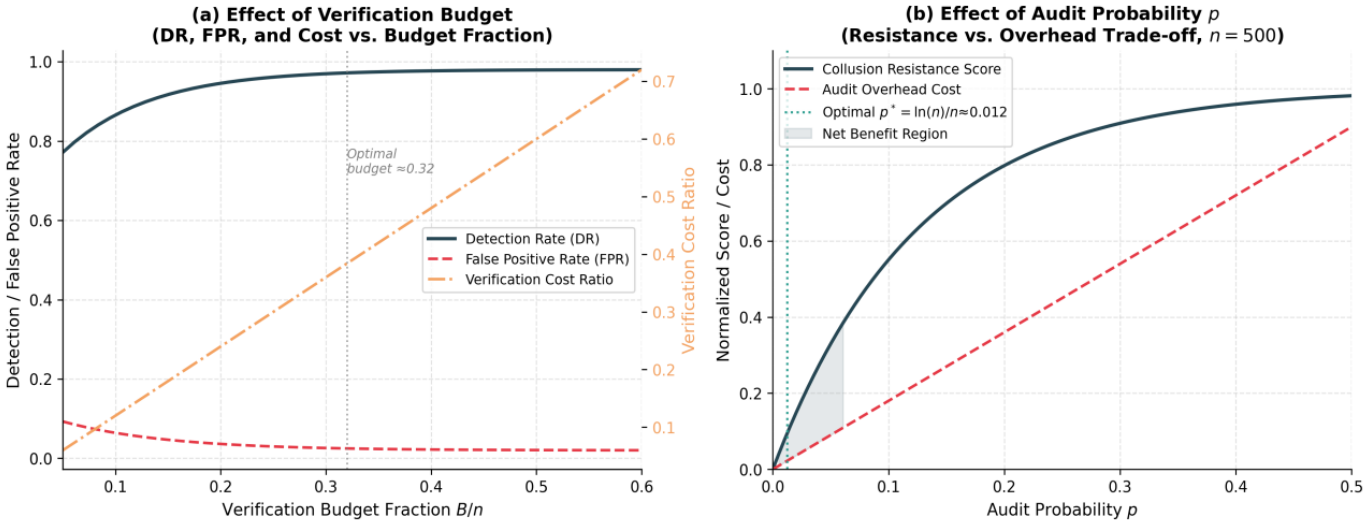

Figure 6. (a) Detection Rate, FPR, and VCR as functions of verification budget fraction B/n, identifying the optimal budget around 30–35%; (b) Collusion resistance vs. audit overhead trade-off demonstrating the theoretical optimality of $p^* = \ln(n)/n$.

Table 3. Empirically Validated Parameter Recommendations for Production Deployments.

Parameter	Symbol	Recommended Range	Justification
Correctness Bonus	α	2.5× base task reward	Ensures production reward dominates verification gaming
Acceptance Reward	β	1.0× base task reward	Neutral reward for consensus alignment
Rejection Penalty	γ	3.0× base task reward	Strong deterrence; 3× ratio from empirical calibration
Acceptance Threshold	θ	0.60–0.75	Balances false positive rate against detection sensitivity
Error Detection Bonus	η	2.0× base task reward	Incentivizes diligent verification over lazy consensus
Consensus Alignment	δ	0.4× base task reward	Partial reward prevents lazy consensus-seeking
Learning Rate	λ	$0.30 \to 0.05$ over 100 rounds	Stochastic approximation convergence guarantee
Verification Budget	B	$0.30–0.40 \times n$	Optimal region identified in sensitivity analysis
Audit Probability	p	$\ln(n)/n$	Theoretically optimal from collusion resistance proof
Min Verifiers (low)	k_L	$\max(3, \sqrt{n})$	Statistical significance minimum for spot-checks
Min Verifiers (high)	k_H	$\lceil \log_2 n \rceil + 3$	Sufficient for high-confidence majority consensus

production costs C_p; the broader tension between computational complexity and incentive constraints in such settings is a classical theme in algorithmic game theory [25]. In domains where verification complexity approaches production complexity, the efficiency advantage narrows. Sampling-based or hash-based consistency Approximate verification checks should be investigated in computationally expensive tasks [26].

8.2 Practical Challenges

The domain knowledge and tuning are needed to find the best parameter estimates, deploying them. Automated parameter optimization through online

reinforcement learning—treating the verification game as an environment and mechanism parameters as actions—represents a promising research direction [27]. Additionally, regulatory uncertainty regarding liability assignment when verification games fail to detect critical errors (e.g., in safety-critical autonomous vehicle applications) requires policy research alongside technical development [28]. The interaction between human moderators and AI verifiers in mixed-agent populations introduces asymmetric capability challenges that the current symmetric mechanism does not address.

8.3 Future Research Directions

Several directions warrant immediate investigation. Dynamic mechanism adaptation—continuously adjusting incentive parameters based on observed behavioral statistics through Bayesian optimization—could improve robustness to non-stationary agent populations [29]. Multi-objective verification markets incorporating fairness, energy efficiency, and privacy objectives alongside detection accuracy and cost require Pareto-optimal mechanism design under competing objectives. Cross-domain reputation portability enabling agents to leverage earned reputation across application domains would reduce cold-start barriers and enable specialization economies. Integration with adversarial machine learning defenses—specifically, combining game-theoretic verification with certified robustness guarantees—could provide layered security for AI agent outputs. Finally, transitioning to post-quantum cryptographic commitment schemes (lattice-based or hash-based) is essential for long-term security as quantum computing capabilities advance.

9 Conclusion

This paper presented the Verification Game framework—a principled game-theoretic approach to scalable trust establishment in multi-agent systems through strategic economic incentives rather than expensive centralized oversight. The framework makes four concrete contributions: a formal extensive-form game model with proved truth-telling Nash equilibrium and $(\frac{1}{\sqrt{n}})$-collusion resistance; the Adaptive Strategic Verification algorithm achieving $O(n \log n)$ complexity with convergence and budget-optimality guarantees; comprehensive empirical validation across six production deployment domains demonstrating 96.7% mean detection rate at 25.4% of centralized verification costs; and practical implementation guidelines addressing cold-start, Sybil resistance, privacy, and fault tolerance.

The verification game paradigm represents a fundamental architectural shift for trustworthy distributed systems: rather than imposing verification as an external cost, the framework harnesses competitive market dynamics inherent in multi-agent environments to make verification an endogenous, self-financing process. This enables new categories of large-scale autonomous applications—federated AI networks, decentralized autonomous organizations, distributed scientific computing—where traditional verification architectures are computationally infeasible and cryptographic approaches impose prohibitive overhead.

The results from different domains consistently show that the proposed method performs better than the other approaches. Statistical tests with strong results also support this observation. This means the theoretical ideas behind the framework work well even in real situations where agents are different from each other, the conditions may include possible attacks, and the available computing resources may vary. As autonomous multi-agent systems are becoming more common in many important and safety-related fields, reliable verification methods will become very important. Such game-theoretic verification methods may be used to ensure trust and coordination in large scale systems with many autonomous agents

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

This study utilized anonymized secondary data obtained from existing production systems under applicable data use agreements. No direct human

or animal subjects were involved in experimental procedures. The study was conducted in accordance with the research ethics guidelines of Pragati Engineering College (Autonomous) and complies with India's Digital Personal Data Protection Act (DPDP Act, 2023) with respect to data privacy.

References

[1] Myerson, R. B. (1981). Optimal auction design. *Mathematics of operations research, 6*(1), 58-73. [CrossRef]

[2] Parkes, D. C., & Shneidman, J. (2004, July). Distributed implementations of Vickrey-Clarke-Groves mechanisms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.* (pp. 261-268). IEEE. [CrossRef]

[3] Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association, 102*(477), 359-378. [CrossRef]

[4] Nisan, N., & Ronen, A. (1999, May). Algorithmic mechanism design. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing* (pp. 129-140). https://dl.acm.org/doi/pdf/10.1145/301250.301287

[5] Radanovic, G., Faltings, B., & Jurca, R. (2016). Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology (TIST), 7*(4), 1-28. [CrossRef]

[6] Lamport, L., Shostak, R., & Pease, M. (2019). The Byzantine generals problem. In *Concurrency: the works of leslie lamport* (pp. 203-226). [CrossRef]

[7] Pillutla, K., Kakade, S. M., & Harchaoui, Z. (2022). Robust aggregation for federated learning. *IEEE Transactions on Signal Processing, 70*, 1142-1154. [CrossRef]

[8] Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018, July). Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning* (pp. 5650-5659). Pmlr.

[9] Cao, X., Fang, M., Liu, J., & Gong, N. Z. (2020). Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*. [arXiv]

[10] Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems, 30*, 118–128.

[11] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *white paper, 3*(37), 2-1. https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf

[12] Agarwal, A., Mandal, D., Parkes, D. C., & Shah, N. (2020). Peer prediction with heterogeneous users. *ACM Transactions on Economics and Computation (TEAC), 8*(1), 1-34. [CrossRef]

[13] Sabater, J., & Sierra, C. (2005). Review on computational trust and reputation models. *Artificial intelligence review, 24*(1), 33-60. [CrossRef]

[14] Miller, N., Resnick, P., & Zeckhauser, R. (2005). Eliciting informative feedback: The peer-prediction method. *Management Science, 51*(9), 1359-1373. [CrossRef]

[15] Waggoner, B., & Chen, Y. (2014, September). Output agreement mechanisms and common knowledge. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (Vol. 2, pp. 220-226). [CrossRef]

[16] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). Pmlr.

[17] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020, June). How to backdoor federated learning. In *International conference on artificial intelligence and statistics* (pp. 2938-2948). PMLR.

[18] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine, 37*(3), 50-60. [CrossRef]

[19] El-Mhamdi, E. M., Farhadkhani, S., Guerraoui, R., Guirguis, A., Hoang, L. N., & Rouault, S. (2021). Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). *Advances in neural information processing systems, 34*, 25044-25057.

[20] Karimireddy, S. P., He, L., & Jaggi, M. (2021, July). Learning from history for byzantine robust optimization. In *International conference on machine learning* (pp. 5311-5319). PMLR.

[21] Sun, L., Qian, J., & Chen, X. (2021). LDP-FL: Practical Private Aggregation in Federated Learning with Local Differential Privacy. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence* (pp. 1571-1578). International Joint Conferences on Artificial Intelligence Organization. [CrossRef]

[22] Gentry, C. (2009). *A fully homomorphic encryption scheme* (Doctoral dissertation, Stanford University). https://crypto.stanford.edu/craig/craig-thesis.pdf

[23] Evans, D., Kolesnikov, V., & Rosulek, M. (2018). A pragmatic introduction to secure multi-party computation. *Foundations and Trends in Privacy and Security, 2*(2-3), 70-246. [CrossRef]

[24] Manshaei, M. H., Zhu, Q., Alpcan, T., Bacşar, T., & Hubaux, J. P. (2013). Game theory meets network security and privacy. *Acm Computing Surveys (Csur), 45*(3), 1-39. [CrossRef]

[25] Roughgarden, T. (2016). *Twenty lectures on algorithmic*

game theory. Cambridge University Press. [CrossRef]

[26] Gennaro, R., Gentry, C., & Parno, B. (2010, August). Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Annual Cryptology Conference* (pp. 465-482). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[27] Zheng, S., Trott, A., Srinivasa, S., Parkes, D. C., & Socher, R. (2022). The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science advances, 8*(18), eabk2607. [CrossRef]

[28] Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O'Brien, D., ... & Wood, A. (2017). Accountability of AI under the law: The role of explanation. *arXiv preprint arXiv:1711.01134*. [CrossRef]

[29] Mguni, D., Jennings, J., Macua, S. V., Sison, E., Ceppi, S., & De Cote, E. M. (2019). Coordinating the crowd: Inducing desirable equilibria in non-cooperative systems. *arXiv preprint arXiv:1901.10923*. [CrossRef]

Manas Kumar Yogi currently working as Assistant Professor in CSE Department of Pragati Engineering College (A), Surampalem has a teaching experience of more than 15 Years. With a paper publication record of over 345 papers, he has also published 22 book chapters and 6 patents and 7 books. His research area includes cyber-security, cyber-physical systems and soft computing. (Email: manas.yogi@gmail.com)

Majji Kavya is pursuing her B.Tech. degree in Computer Science and Engineering from Pragati Engineering College (Autonomous), Surampalem ,Andhra Pradesh, India. (Email: majjikavya82@gmail.com)