INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY TRENDS

Vol. 06 No. 01 (2026)

e. ISSN: 2409-5222

https://ijictt.ictt.info

Dynamic Trust Decay Modeling for Containment of Memory Poisoning in Collaborative Agent Networks

Oliver Bennett1, Daniel Hughes2, Emily Carter3*

Department of Computer Science, The University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom

Corresponding author: *emily.carter@manchester.ac.uk*

Abstract: *Persistent poisoning may exploit static trust assumptions among agents. This study introduces a dynamic trust decay model in which agent credibility scores decrease proportionally to detected inconsistency in memory contributions. Trust scores are updated using a nonlinear decay function based on deviation magnitude and frequency. Agents with trust below a predefined threshold are temporarily restricted from contributing shared memory. Simulations were conducted on 120-agent coordination tasks with varying poisoning intensities. The trust decay mechanism reduced cross-agent contamination spread by 57.2% and improved overall task success rate from 62.8% to 84.5% under a 20% poisoning scenario. System stability was restored within 18 update cycles compared to 41 cycles in static-trust settings. Dynamic trust modeling effectively limits poisoning persistence and accelerates recovery in collaborative environments.*

Keywords: *Multi-agent systems; trust modeling; memory poisoning containment; credibility decay; collaborative networks; resilience mechanisms*

1. INTRODUCTION

Collaborative multi-agent systems are widely used in distributed control, task coordination, and agent-based decision-making environments. By exchanging local observations, historical records, and intermediate reasoning results, agents can form a broader view of the environment and improve collective task performance. This collaborative design, however, also creates a structural vulnerability: once unreliable or malicious information is injected into shared memory, it can persist across multiple interaction rounds, influence later decisions, and propagate through the network during continued cooperation [1,2]. This persistent propagation behavior has been explicitly analyzed in recent work, which shows that poisoned memory can spread across agents and remain effective over multiple interaction cycles [3]. Recent studies have shown that memory poisoning is not limited to an isolated local disturbance. In collaborative settings, harmful content may be repeatedly retrieved, reused, and redistributed, which allows its influence to accumulate over time and spread beyond the initially affected agent [4]. This persistence makes memory integrity a central issue in secure multi-agent collaboration, especially in systems that rely on long-term memory, iterative planning, or repeated information exchange [5]. Trust modeling has long been regarded as an effective mechanism for handling uncertainty in multi-agent interaction. Trust-aware coordination can help agents reduce dependence on unreliable neighbors, improve consensus quality, and support more stable cooperation under imperfect information [6,7]. In many existing studies, trust is treated as a relatively stable property or is updated with limited sensitivity to behavioral change. Such designs are often insufficient for collaborative environments in which agent reliability may shift rapidly due to faults, adversarial manipulation, or repeated abnormal

memory contributions. At the same time, recent work on agent security has identified memory poisoning as a major threat in both memory-based and LLM-driven agent systems [8,9]. Harmful content can be inserted into long-term memory and later retrieved during reasoning, planning, or coordination, while remaining difficult to distinguish from normal information during routine interaction [10,11]. In collaborative networks, this problem becomes more severe because contaminated memory does not remain confined to one agent. Once shared, it may affect the behavior of multiple agents and degrade overall system performance. Related findings have also appeared in studies of adversarial communication in multi-agent systems [12]. Even a small number of corrupted messages or compromised agents can reduce coordination quality, distort decision trajectories, and trigger broader instability in the network [13,14]. Although these studies provide useful insight into the security risks of cooperative systems, much of the existing literature still emphasizes short-term communication disturbances rather than persistent memory contamination. Compared with one-step or short-horizon attacks, poisoned memory can remain active long after the initial intrusion and continue to influence downstream interactions. This long-lasting effect changes the nature of the defense problem: the challenge is no longer limited to filtering incoming messages, but extends to identifying unreliable memory sources, controlling contamination spread, and restoring collaboration quality after disruption [15,16]. Current research still leaves several important issues insufficiently addressed. Many poisoning studies focus on single-agent settings and therefore do not fully characterize how harmful memory propagates across interconnected agents in collaborative networks [17]. Trust modeling and memory defense are also often studied separately, although repeated inconsistency in shared memory contributions is a direct indicator of declining agent credibility. In addition, many existing experiments are conducted in relatively small or task-specific environments, which makes it difficult to assess contamination control, recovery behavior, and coordination stability under broader collaborative conditions. As a result, there remains a need for a unified mechanism that can interpret abnormal memory updates through trust dynamics and use this information to limit the influence of poisoned contributors before contamination spreads further. Against this background, the present study develops a dynamic trust decay model for containing memory poisoning in collaborative agent networks. The proposed method adjusts each agent's trust score according to the magnitude and recurrence of inconsistency in its shared memory contributions, so that abnormal behavior is not treated as an isolated event but as part of a changing reliability pattern. A nonlinear decay mechanism is introduced to accelerate trust reduction under repeated suspicious updates, allowing the system to respond more quickly to persistent contamination than conventional fixed or slowly updated trust schemes. Agents whose trust scores fall below a defined threshold are temporarily prevented from contributing to shared memory, thereby limiting the diffusion of unreliable information across the network. The significance of this study lies in linking trust adaptation with memory protection in a unified defense framework. Rather than viewing poisoning only as a data-quality problem, the proposed approach interprets it as a dynamic credibility problem embedded in collaborative interaction. This perspective enables the system to suppress contamination spread, improve recovery speed after attack, and preserve task performance under hostile conditions. The study also contributes to the understanding of secure coordination in memory-dependent multi-agent systems by examining how trust-aware containment can support more stable collaboration when memory integrity is threatened. These findings are expected to provide practical support for the design of safer collaborative agents in distributed control, cooperative planning, and other networked intelligent environments.

2.Materials and Methods

2.1 Sample and Task Environment

The study was conducted on a simulated collaborative agent network for large-scale coordination tasks. Each experiment included 120 agents. All agents operated in the same task environment and followed the same action limits, memory capacity, and update interval. The dataset consisted of memory contribution records collected during repeated coordination

cycles. Each record included the agent ID, local memory update, deviation score, trust score, and task feedback at that cycle. To test poisoning effects, part of the agent population was set as compromised under controlled attack conditions. These agents injected inconsistent memory contributions at preset poisoning levels. The benchmark included both normal and adversarial settings so that the behavior of the trust decay method could be examined under stable and disturbed conditions. All agents were initialized with the same parameter settings. This setup ensured that the observed differences were mainly related to trust control rather than unequal initial conditions.

2.2 Experimental Design and Control Setup

A comparative experiment was used to examine the effect of dynamic trust decay on containment of memory poisoning. In the experimental group, each agent was assigned a trust score that changed over time according to the size and frequency of inconsistency in its memory contributions. Agents whose trust scores fell below a preset threshold were temporarily prevented from contributing to shared memory. In the control group, a static-trust scheme was used, in which all agents kept fixed credibility weights during the task process unless they were manually removed. Both groups were tested under the same poisoning levels and coordination tasks. This design allowed direct comparison of contamination spread, task success rate, and recovery speed. The static-trust setting was used as the baseline because it reflects a common assumption in collaborative systems. The dynamic trust design was adopted because agent reliability may change across update cycles in adversarial settings.

2.3 Measurement Procedures and Quality Control

At each update cycle, every agent submitted a local memory contribution to the shared coordination process. The system then measured the deviation of each contribution from the current group pattern and used this information to update the agent's trust score in the experimental setting. Three main outcome variables were recorded: cross-agent contamination spread, overall task success rate, and recovery time after poisoning disturbance. Contamination spread was measured as the proportion of normal agents whose shared memory states were affected by poisoned contributions. Task success rate was calculated as the percentage of coordination runs that reached the preset task goal. Recovery time was defined as the number of update cycles required for system performance to return to a stable level after poisoning began. To ensure data quality, all experiments were run under the same hardware setting, random seed range, task horizon, and memory update schedule. Each condition was repeated several times, and the mean values were used in the final analysis. Log records were checked for missing values, execution faults, and abnormal interruptions unrelated to poisoning. Runs affected by technical failure were removed before statistical analysis.

2.4 Data Processing and Model Formulation

All outputs were grouped by trust strategy and poisoning level before analysis. Mean values and standard deviations were first calculated for each index across repeated trials. The trust score of agent i at update cycle t+1 was defined as

$$T_i^{(t+1)}=T_i^{(t)}\exp\left(-\alpha d_i^{(t)}-\beta f_i^{(t)}\right)$$

where $T_i^{(t)}$ is the trust score at cycle t, $d_i^{(t)}$ is the deviation size of the current memory contribution, $f_i^{(t)}$ is the recent frequency of inconsistent contributions, and α and β are decay coefficients. This form was used so that trust dropped faster when both deviation size and repetition increased. In addition, contamination spread was calculated as

$$C=\frac{N_c}{N_n}\times 100\%$$

where N_c is the number of normal agents affected by poisoned memory propagation and N_n is the total number of normal agents in the network. All variables were processed with the same pipeline so that the final comparison reflected differences in trust modeling rather than differences in data handling.

2.5 Statistical Analysis and Evaluation Criteria

The proposed method was evaluated from three aspects: containment effect, task performance, and recovery efficiency. The containment effect was assessed by comparing contamination spread between the dynamic-trust model and the static-trust baseline. Task performance was assessed by the overall success rate of coordination tasks under different poisoning levels. Recovery efficiency was assessed by the number of update cycles needed to restore stable operation after disturbance. Mean values across repeated runs were used as the main results, and variation across runs was used to examine consistency. A lower contamination ratio indicated better control of poisoning spread. A higher success rate indicated stronger resistance to poisoned memory. A shorter recovery period indicated faster restoration of system stability. The proposed method was considered effective when it reduced cross-agent contamination, improved task completion, and shortened recovery time under the same poisoning condition.

3.Results and Discussion

3.1 Containment of poisoning spread

The dynamic trust decay model clearly reduced the spread of poisoned memory in the agent network. Under the 20% poisoning setting, cross-agent contamination was 57.2% lower than that of the static-trust baseline. This result suggests that fixed trust assumptions allowed harmful updates to remain active for too long. In the proposed method, repeated inconsistency caused trust to fall quickly, so suspicious agents were restricted before their memory contributions reached a larger part of the network. As shown in Fig. 1, the main advantage of the method was early containment rather than delayed correction. This result is consistent with recent trust studies showing that adaptive credibility control responds more effectively to behavior change than fixed trust settings, especially when malicious behavior appears gradually [18,19].

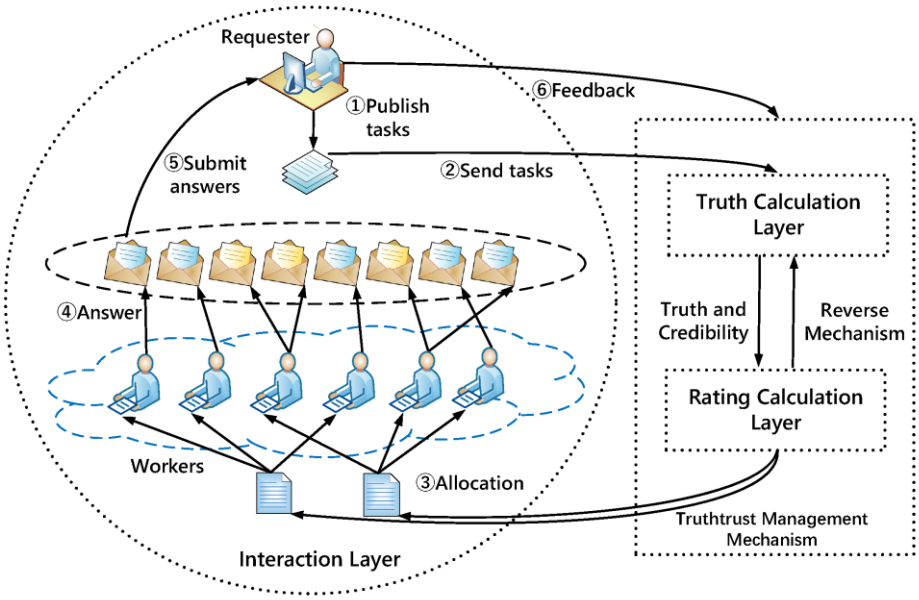


Figure 1: Cross-agent contamination under dynamic trust decay and static trust.

3.2 Improvement in task success under attack

The proposed method also improved the overall success rate of collaborative tasks. Under the same poisoning condition, task success increased from 62.8% in the static-trust setting to 84.5% with dynamic trust decay. This result indicates that poisoning affected not only local memory quality but also coordination at the system level. When unreliable agents continued to provide shared memory, the error entered later update rounds and reduced joint task completion. By reducing the influence of agents with repeated abnormal updates, the proposed method helped preserve useful shared information and maintain stable cooperation. Similar results have been reported in trust-aware and attack-resistant coordination studies, where

selective weighting of information sources improved system performance under unreliable interaction [20,21].

3.3 Recovery speed and system stability

Another important result is the shorter recovery time after poisoning began. In the static-trust setting, the system needed 41 update cycles to return to a stable state. In contrast, the proposed method restored stability within 18 cycles. This difference shows that the dynamic trust mechanism not only blocked harmful influence but also helped the system recover more quickly after disturbance. In practical terms, once trust fell below the threshold, repeated abnormal contributors lost access to shared memory. This reduced further contamination and allowed the remaining agents to rebuild a reliable coordination pattern. As shown in Fig. 2, this recovery process depended on continuous trust revision rather than one-time detection. Earlier work on adaptive trust estimation reported a similar pattern: when agent behavior changes over time, trust must be updated in a timely way, or the system remains affected by outdated credibility assumptions for too many rounds [22,23].

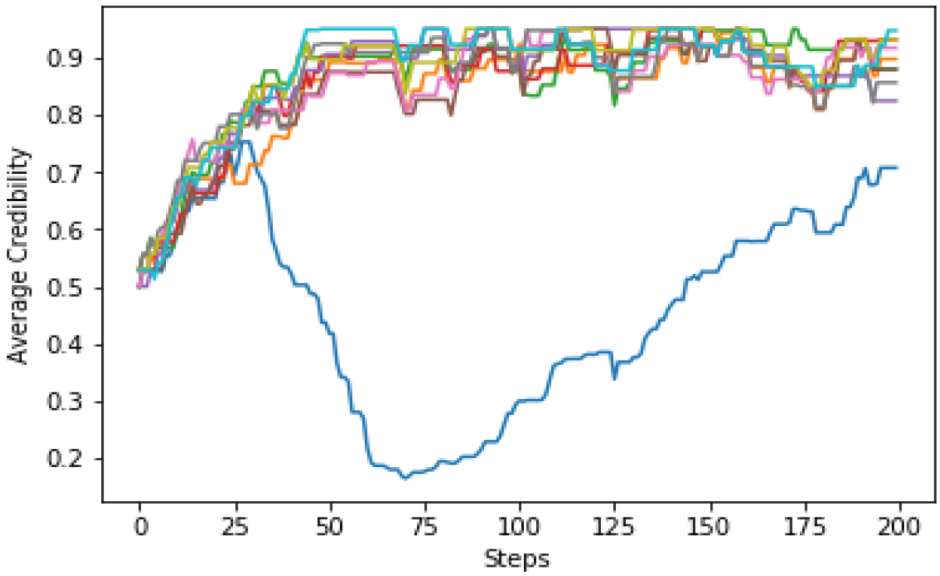


Figure 2: System recovery under dynamic trust updating in collaborative agent networks.

3.4 Comparison with earlier studies and current limits

Compared with earlier studies, the present work adds value in two ways. First, many recent poisoning studies focus on federated model updates or short-term message attacks, whereas this study examines persistent memory contamination in collaborative agent networks. Second, several trust models estimate agent reliability, but they are not directly linked to the control of poisoned shared memory. The present framework connects trust decay with inconsistency in memory contribution, which makes the control rule more suitable for repeated collaboration under attack. However, some limits remain. The experiments were carried out on 120-agent coordination tasks with preset poisoning levels, and the attack model focused on inconsistent memory injection rather than more complex collusive strategies. In addition, the threshold-based restriction rule may become less effective if malicious agents closely imitate normal contribution patterns. Future work should examine denser attack settings, adaptive thresholds, and topology-aware trust control in larger collaborative networks.

4.Conclusion

This study developed a dynamic trust decay model to limit memory poisoning in collaborative agent networks. The method updated each agent's trust score based on the size and frequency of inconsistency in its memory contributions, and agents with low trust were temporarily removed from shared memory exchange. The results showed that this design reduced cross-agent contamination by 57.2%, increased task success from 62.8% to 84.5% under a 20% poisoning condition, and shortened recovery time from 41 to 18 update cycles. These results suggest that fixed trust settings are not suitable for collaborative environments in which agent

behavior may change over time. The main contribution of this work is the direct connection between trust control and memory integrity, so that poisoning containment becomes a continuous adjustment process instead of a one-time filtering step. This gives the study practical value for distributed coordination, cooperative decision systems, and other agent networks that depend on repeated memory exchange in uncertain or hostile settings. The proposed framework also has potential for large collaborative systems where both rapid containment and stable recovery are important. However, some limits remain. The experiments were carried out on a 120-agent benchmark with preset poisoning levels, and the attack setting focused on inconsistent memory injection rather than more complex collusive behavior. In addition, the threshold-based restriction rule may become less effective when malicious agents closely copy normal contribution patterns. Future work should test larger networks, more complex attack settings, and more flexible trust update rules to further improve robustness and practical value.

References

Bai, W., Wu, Q., Wu, K., & Lu, K. (2024). Exploring the Influence of Prompts in LLMs for Security-Related Tasks. In Workshop on Artificial Intelligence System with Confidential Computing (AISCC 2024)(San Diego, CA). USA. https://dx. doi. org/10.14722/aiscc.

Batista, A. D. S., & Dos Santos, A. L. (2024). A survey on resilience in information sharing on networks: Taxonomy and applied techniques. ACM Computing Surveys, 56(12), 1-36.

Liu, H., Xu, D., Ma, Q., Xu, S., & Qiu, D. (2026). Memory Poisoning Propagation and Repair Mechanism in Multi-Agent Collaborative Environments.

Agarwal, D., Shahid, F., & Vashistha, A. (2024). Conversational agents to facilitate deliberation on harmful content in whatsapp groups. Proceedings of the ACM on human-computer interaction, 8(CSCW2), 1-32.

Krishnan, N. (2025). Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. arXiv preprint arXiv:2504.21030.

Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.

Chen, H., Li, J., Ma, X., & Mao, Y. (2025, June). Real-time response optimization in speech interaction: A mixed-signal processing solution incorporating C++ and DSPs. In 2025 7th International Conference on Artificial Intelligence Technologies and Applications (ICAITA) (pp. 110-114). IEEE.

Zambare, P., Thanikella, V. N., & Liu, Y. (2025). Securing agentic ai: Threat modeling and risk analysis for network monitoring agentic ai system. arXiv preprint arXiv:2508.10043.

Rahman, M. A., Francia, G., Shahriar, H., Cuzzocrea, A., Mohamed, A., Khan, M. U., & Ahamed, S. I. (2025, December). A Survey of Large Language Models (LLMs) for Cybersecurity: Opportunities and Directions. In 2025 IEEE International Conference on Big Data (BigData) (pp. 4333-4342). IEEE.

Li, T., Liu, S., Hong, E., & Xia, J. (2025). Human Resource Optimization in the Hospitality Industry Big Data Forecasting and Cross-Cultural Engagement.

Maragheh, R. Y., & Deldjoo, Y. (2025). The future is agentic: Definitions, perspectives, and open challenges of multi-agent recommender systems. arXiv preprint arXiv:2507.02097.

Qiu, D., Xu, D., & Yue, L. (2025, December). Reinforcement Learning-Augmented LLM Agents for Collaborative Decision Making and Performance Optimization. In 2025 7th International Conference on Frontier Technologies of Information and Computer (ICFTIC) (pp. 1337-1342). IEEE.

Habibullah, S. M., & Dhanekula, A. (2023). Blockchain-Orchestrated Cyber-Physical Supply Chain Networks with Byzantine Fault Tolerance For Manufacturing Robustness. Journal of Sustainable Development and Policy, 2(03), 34-72.

Gu, X., Yang, J., Tian, X., & Liu, M. (2025). Research on the Construction of a Human-Machine Collaborative Anti-Money Laundering System and Its Efficiency and Accuracy Enhancement in Suspicious Transaction Identification.

Batista, A. D. S., & Dos Santos, A. L. (2024). A survey on resilience in information sharing on networks: Taxonomy and applied techniques. ACM Computing Surveys, 56(12), 1-36.

Yang, Y., Leuze, C., Hargreaves, B., Daniel, B., & Baik, F. (2025). EasyREG: Easy Depth-Based Markerless Registration and Tracking using Augmented Reality Device for Surgical Guidance. arXiv preprint arXiv:2504.09498.

de Witt, C. S. (2025). Open challenges in multi-agent security: Towards secure systems of interacting ai agents. arXiv preprint arXiv:2505.02077.

Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.

Bonfanti, R. C., Ruggieri, S., & Schimmenti, A. (2024). Psychological trust dynamics in climate change adaptation decision-making processes: a literature review. Sustainability, 16(10), 3984.

Ma, Q., Yue, L., Xu, S., Shi, Y., & Liu, H. (2026). Web Agent Agentic Reinforcement Learning Decision Model Under Multi-Cost and Failure Risk Constraints.

Sameer, M. M. (2025). A Context-Aware Hybrid Trust Management Model for the Internet of Vehicles.

Qiu, Y. (2024). Estimation of tail risk measures in finance: Approaches to extreme value mixture modeling. arXiv preprint arXiv:2407.05933.

Lygizou, Z., & Kalles, D. (2025). A biologically inspired trust model for open multi-agent systems that is resilient to rapid performance fluctuations. Applied Sciences, 15(11), 6125.