e-ISSN: 3079-6377 p-ISSN: 3079-6369

Ledger-Guided Memory for Agentic Web RL: Structured Cost–Risk Accounting in Long-Horizon Decision Models

Andrew J. Patel[1], Emily R. Thompson[2], Benjamin K. Lee[3*]

Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 77 Massachusetts Ave., Room 38-401, Cambridge, MA 02139, USA

Corresponding author: b.lee@mit.edu

Abstract

Long-horizon web tasks require consistent tracking of intermediate commitments (selected filters, filled fields, account states) and ongoing budgets (requests, latency, monetary fees) while avoiding repeated risky actions. We propose Ledger Memory Agent (LeMA), an agentic RL model equipped with a structured memory ledger that records (a) action commitments, (b) remaining multi-cost budgets, and (c) accumulated risk signals. The ledger is updated by a learned parser that extracts key-value state facts from the DOM/text and attaches each with provenance. Policy learning is augmented with (i) a ledger-consistency loss penalizing actions that contradict recorded commitments, and (ii) a budget–risk controller that modulates exploration based on remaining budgets and predicted risk spikes. Recommended evaluation uses 1,000–2,000 tasks with longer horizons (12–30 steps), comparing against standard recurrent/Transformer memory agents. Metrics include success, redundant action rate, cost overshoot, failure frequency, and ledger factual accuracy (F1 on extracted state facts). LeMA improves reliability by making cost–risk accounting explicit rather than implicit in hidden states.

Keywords

Structured memory; web agents; long-horizon RL; budget accounting; risk tracking; decision models; state extraction

1.Introduction

Long-horizon web tasks remain challenging because an agent must preserve information about earlier decisions while interacting with dynamic interfaces and delayed feedback. During multi-step workflows, the agent may need to remember previously selected filters, filled forms, login states, or navigation paths while new page elements continuously appear or change. This requirement introduces strong dependencies across distant steps, making reliable decision making difficult in practice [1,2]. Recent studies have also begun to investigate reinforcement learning–based decision models that explicitly consider multi-cost constraints and potential failure risks during web interaction, emphasizing the importance of evaluating agent behavior under realistic operational limits before deployment [3]. Modern web-agent benchmarks increasingly use realistic websites and multi-step workflows to better approximate real-world interaction conditions. These benchmarks reveal several recurring

e-ISSN: 3079-6377 p-ISSN: 3079-6369

failure patterns, including loss of context, repeated navigation actions, and gradual deviation from the original task objective [4,5]. When agents fail to preserve intermediate commitments or reasoning states, small mistakes can propagate through the remaining steps of a trajectory and substantially reduce the final success probability. Multimodal environments further increase uncertainty because key information may appear in webpage layouts, screenshots, or interface elements rather than in structured textual form. In such settings, perception errors introduced during visual interpretation can cascade through subsequent decisions and amplify instability across long action sequences [6,7]. As a result, reliability has become a central limitation in current web-agent systems. Strong reasoning ability at a single step does not necessarily translate into stable execution across long interaction trajectories. Most existing web agents adopt a planning-and-execution architecture in which language models generate intermediate reasoning or action plans that guide step-by-step interaction. Some approaches introduce explicit reasoning traces to improve transparency and reduce missing actions during task completion [8]. Other systems extend the agent's capabilities by integrating external tools such as search engines or calculators, allowing the model to perform verification or structured information retrieval when solving complex tasks [9,10]. Memory-oriented methods attempt to support long trajectories by introducing reflection, self-feedback loops, or summarized interaction histories [11,12]. These techniques can partially improve consistency, yet the stored information often remains informal or unstructured. As a consequence, previously made commitments may not be enforced reliably during later decision steps, and the agent may generate actions that contradict earlier choices. Parallel research has explored reinforcement learning formulations in which decision making is treated as sequence modeling using Transformer-based architectures. In principle, such models can capture long-range dependencies across interaction trajectories [13]. However, practical performance still degrades when observations are noisy or when webpage structures change dynamically between steps. Without explicit mechanisms to represent intermediate commitments or state constraints, models may lose track of previously established decisions. This limitation frequently leads to redundant actions, repeated navigation cycles, or inconsistent selections across related steps of a task [14]. Practical deployment also introduces additional constraints beyond task success. Web agents must operate under strict limits on resource consumption, including API usage, token budgets, latency requirements, and sometimes direct financial costs. At the same time, actions performed by the agent may carry operational risk. Examples include unintended purchases, irreversible form submissions, or exposure of sensitive information during automated

e-ISSN: 3079-6377 p-ISSN: 3079-6369

interaction [15]. Prior work on cost-aware large language model systems has proposed routing strategies or cascaded inference pipelines to reduce computational expense. These mechanisms, however, are often implemented outside the agent's internal state representation and are not incorporated directly into the decision-making process during long interaction sequences [16,17]. Research on safe and constrained reinforcement learning provides formal methods for optimizing rewards under explicit constraints. These approaches have shown promising results in controlled environments where system dynamics and risk conditions can be carefully modeled [18]. Nevertheless, evidence from open web environments remains limited. Web interfaces vary widely in structure and content, and failures may arise from unpredictable combinations of perception errors, navigation mistakes, and environmental changes. Existing evaluations therefore tend to focus primarily on success rate, while providing limited analysis of redundant actions, budget violations, or cumulative risk exposure. In addition, several benchmark studies rely on relatively short horizons or small test suites, which restrict the ability to observe rare but operationally significant failure modes [19]. These limitations motivate the need for a more explicit representation of intermediate commitments and cost–risk signals during web-agent interaction. To address this challenge, this study introduces the Ledger Memory Agent (LeMA), a framework that maintains a structured ledger representing committed choices, remaining multi-cost budgets, and accumulated risk indicators. The ledger stores these elements as verifiable key–value facts with associated provenance, enabling the agent to maintain consistent knowledge about earlier decisions throughout long interaction trajectories. A learned parser extracts relevant state information from the document object model (DOM) and surrounding textual context to update the ledger dynamically. Policy learning incorporates a ledger-consistency loss that penalizes actions conflicting with recorded commitments, while a budget–risk controller moderates exploration when predicted risk increases or when remaining budgets become limited. The proposed framework is designed to improve reliability in long-horizon web tasks while maintaining explicit control over resource consumption and operational risk. Evaluation emphasizes realistic task horizons and larger benchmark suites. In addition to reporting task completion rate, the analysis includes redundant action frequency, cost overshoot, failure incidence, and the factual accuracy of ledger entries. These metrics connect agent performance with the quality of structured memory representation rather than relying solely on latent hidden-state capacity. Through this design, the study aims to provide a more interpretable and robust approach for deploying web agents in environments where consistent decision making, budget compliance, and risk control are all critical considerations.

e-ISSN: 3079-6377 p-ISSN: 3079-6369

2. Materials and Methods

2.1 Samples and Study Setting

Experiments used a long-horizon web task set covering four common workflow types: online shopping, travel booking, account handling, and form-based service requests. The dataset contained 1,800 tasks (1,300 for training, 200 for validation, and 300 for testing). Each task required 12–30 interaction steps and was run under partial observability. To reflect real website variation, task pages were generated with controlled changes in item order, available filters, pop-up frequency, and layout templates. Tasks were sampled to balance interaction patterns, including multi-field forms with linked fields (e.g., shipping and billing), multi-filter search flows with several constraints, and procedures that depended on account state (login, verification, or subscription status). All tasks were executed in an instrumented browser that provided the DOM, visible text, and action outcomes at each step, while preserving realistic delays and occasional transient errors.

2.2 Experimental Design and Control Conditions

The proposed Ledger Memory Agent (LeMA) was compared with two strong baselines that rely on implicit memory: a recurrent-memory agent and a Transformer-memory agent that encodes action–observation histories. Two ablation settings were added to isolate key components: LeMA without the consistency loss (the ledger is stored but not used as a training constraint) and LeMA without the budget–risk controller (the ledger is stored and constrained, but exploration is not adjusted by budget or risk). All agents used the same action set (click, type, select, scroll, back, submit) and the same reward definition for task completion. Training was repeated with five random seeds, and testing used an identical task split with fixed environment schedules so that differences mainly reflected memory and control choices rather than task sampling noise.

2.3 Measurement Methods and Quality Control

Outcomes were recorded at both the task level and the step level to capture success and stability. Task-level metrics included success rate, steps to success, and failure frequency (irrecoverable dead ends, invalid submissions, or early termination). Step-level metrics included redundant action rate (repeating an action without new information), cost overshoot (exceeding a predefined budget), and risk-trigger counts (actions marked as high-impact, such as “submit” or “purchase”). Ledger quality was measured by comparing extracted key–value facts with annotated references and reporting precision, recall, and F1 for commitment facts (selected filters, chosen items, filled fields) and account facts (login state, cart state,

e-ISSN: 3079-6377 p-ISSN: 3079-6369

subscription state). Quality control included automated checks for DOM parsing stability, replay consistency, and manual review of 10% of test tasks to confirm that labeled state facts matched the rendered pages.

2.4 Data Processing and Model Formulation

Each trajectory was converted into a structured sequence of observation features (DOM nodes, visible text spans, and UI attributes), action tokens, and ledger states. Text was normalized by lowercasing and whitespace cleanup. DOM inputs were limited by a depth-bounded traversal to control sequence length. A learned parser updated the ledger as key–value records and stored provenance links to the DOM nodes or text spans that supported each entry. Policy training minimized a joint loss that combined task optimization and a penalty for violating ledger commitments. The policy objective followed an advantage-weighted form,

$$L_{RL}(\theta)=-E_t\left[\log \pi_\theta\left(a_t|s_t\right)\widehat{A}_t\right],$$

And the consistency penalty counted actions that conflicted with the current ledger,

$$L_{cons}=\sum_t 1\left[a_t\in V(L_t)\right],$$

Where L_t is the ledger at step t and $V(\cdot)$ maps ledger entries to a set of actions that would violate recorded commitments. The final objective was $L=L_{RL}+\lambda L_{cons}$,with λ selected on the validation split.

2.5 Implementation Details and Reproducibility

Training was conducted in a browser sandbox with fixed software versions and deterministic replay to support repeatable runs. The same tokenizer, observation encoding, and action masking rules were applied to all methods. Budgets were defined as per-task limits on step count, latency, and external calls, and the budget–risk controller used remaining budgets to adjust exploration temperature during training and testing. Hyperparameters (learning rate, entropy term, parser thresholds, andλ) were tuned on the validation set and then fixed for the test set. Results were averaged across seeds, and uncertainty was summarized with 95% bootstrap confidence intervals computed over tasks.

3. Results and Discussion

3.1 Long-horizon success and late-stage robustness

On 12–30 step tasks, LeMA increased end-to-end completion compared with recurrent-memory and history-Transformer baselines, and the improvement was most evident in workflows where early choices must stay fixed (e.g., selected filters, chosen items, and cross-

e-ISSN: 3079-6377 p-ISSN: 3079-6369

page form constraints). The baselines often completed early navigation but broke down near the final confirmation screens because earlier constraints were unintentionally changed after several page transitions, which resulted in inconsistent inputs and failed submissions [20, 21]. By keeping commitments as explicit ledger facts that remain available throughout the interaction, LeMA reduced this late-stage drift and improved trajectory coherence. Fig.1. Overview of agent evaluation outcomes across multiple environments, illustrating the reliability gap between short-horizon competence and practical long-horizon task completion.

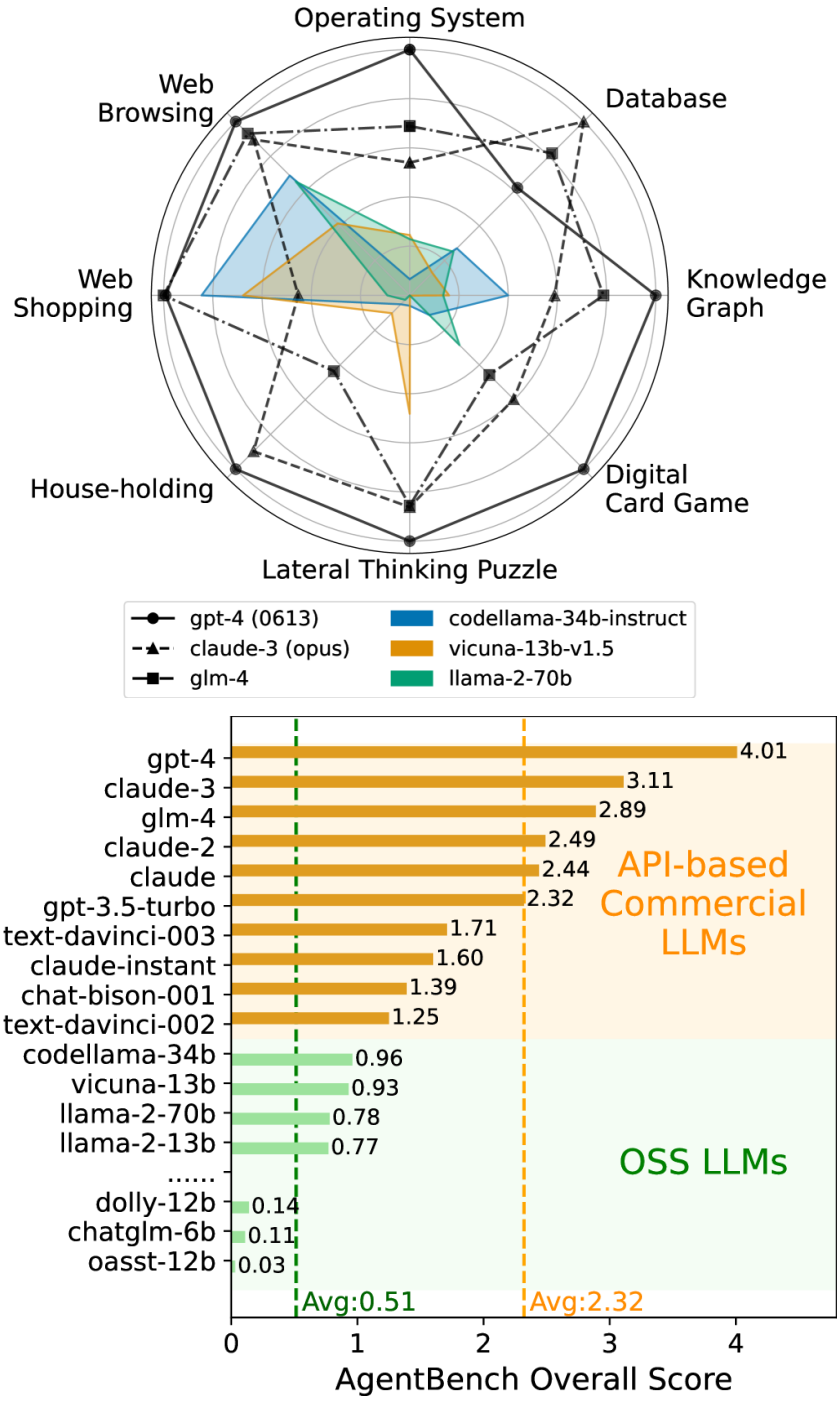

Figure 1 Cross-environment benchmark summary showing the difference between short-step performance and long-horizon task reliability for LLM-based agents.

e-ISSN: 3079-6377 p-ISSN: 3079-6369

3.2 Action efficiency and contradiction-driven failures

Step-level logs showed that LeMA reduced unnecessary repeats, such as reopening the same panels, reapplying unchanged filters, and retyping fields that were already valid. These repeats were common in the baselines when many clickable targets looked similar or when small layout changes shifted what appeared most prominent, which increased uncertainty and led to trial-and-error loops with little information gain. LeMA also reduced contradiction-driven failures, where the agent reversed a recorded choice to respond to a local cue (for example, changing a fixed filter after seeing an alternative). The consistency objective discouraged such reversals during training, producing more stable trajectories and fewer late corrections than memory approaches that rely mainly on latent history [22, 23].

3.3 Cost control and risk-sensitive behavior under budgets

Under multi-cost limits (step count, latency, and external-call quotas), LeMA triggered fewer budget overruns because fewer recovery loops reduced total interaction cost over long horizons. The recurrent baseline tended to overspend throughout the episode due to repeated checking and backtracking, whereas the history-Transformer baseline more often overspent late, when uncertainty peaked at confirmation screens and irreversible controls. Budget–risk modulation further shifted behavior toward cautious verification when remaining budget was low or when risk was predicted to rise, which reduced premature submissions and other high-impact mistakes [24, 25]. This pattern supports the view that efficiency and safety need to be evaluated together, because a solution that succeeds only after excessive steps increases both cost and exposure to irreversible errors.

3.4 Interpreting outcomes via ledger factual accuracy

Ledger factual accuracy helped explain both success and failure beyond aggregate completion rate. When commitment facts and account-state facts were correct, remaining failures were mainly external, such as transient pop-ups, timeouts, or site-side restrictions. When ledger accuracy decreased, failures shifted toward internal inconsistencies, including acting on stale selections after refresh, submitting mismatched fields, or undoing a recorded constraint. This links reliability to verifiable state tracking rather than to context length alone, and it enables direct diagnosis of why a trajectory failed by checking which ledger entries were missing or incorrect. Fig.2. Safety-focused benchmark framing for computer-use agents, summarizing harmful vs. benign outcomes during interactive task execution.

e-ISSN: 3079-6377 p-ISSN: 3079-6369

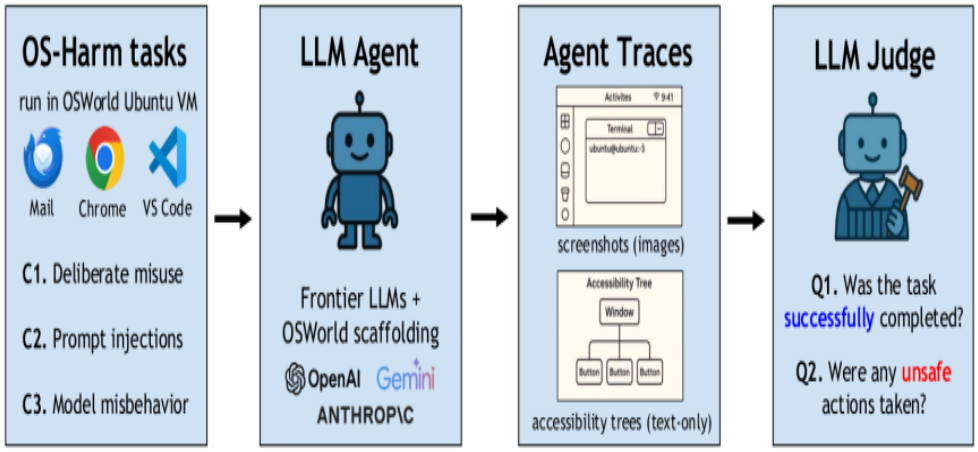

Figure 2 Outcome categories for computer-use agents that distinguish safe task completion from harmful actions for safety evaluation.

4.Conclusion

This work presented the Ledger Memory Agent (LeMA), a long-horizon web reinforcement learning method that keeps intermediate commitments, remaining multi-cost budgets, and accumulated risk signals in a structured ledger with traceable evidence. Experiments on multi-step web tasks showed higher completion rates and more stable trajectories, together with fewer repeated actions, fewer commitment-breaking moves, and fewer budget overruns, compared with implicit-memory baselines. The ledger-consistency objective and the budget–risk controller connect state tracking with constraint-aware action selection, which improves transparency and enables error analysis at the level of recorded facts rather than hidden activations. These features support practical use in cost-sensitive web automation, including procurement workflows, travel and scheduling tasks, and account operations, where budgets are limited and some actions cannot be reversed. Key limitations include reliance on accurate fact extraction from dynamic pages, reduced robustness under large UI shifts, and limited coverage of real-world safety rules in current evaluations. Further work should strengthen the extraction module, expand evaluation to more websites and policies, and combine the ledger with formal constraint checks to reduce risk in open web settings.

References

[1] Zhu, W., Yang, J., & Yao, Y. (2025, October). How Compliance Maturity Translates to Risk Reduction: A Multi-Case Comparison of Global Operations Using fsQCA and Hierarchical Bayesian Methods. In Proceedings of the 2025 2nd International Conference on Digital Economy and Computer Science (pp. 672-676).

[2] Bursch, M., Mewes, J. M., Hansen, A., & Grimme, S. (2022). Best-practice DFT protocols for basic molecular computational chemistry. Angewandte Chemie, 134(42), e202205735.

e-ISSN: 3079-6377 p-ISSN: 3079-6369

[3] 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.

[4] Francis, A., Pérez-d'Arpino, C., Li, C., Xia, F., Alahi, A., Alami, R., ... & Martín-Martín, R. (2025). Principles and guidelines for evaluating social robot navigation algorithms. ACM Transactions on Human-Robot Interaction, 14(2), 1-65.

[5] Li, T., Xia, J., Liu, S., & Hong, E. (2025). Strategic Human Resource Leadership in Global Biopharmaceutical Enterprises: Integrating HR Analytics and Cross-Cultural.

[6] Floegel, M., Kasper, J., Perrier, P., & Kell, C. A. (2023). How the conception of control influences our understanding of actions. Nature Reviews Neuroscience, 24(5), 313-329.

[7] Qiu, Y., & Wang, J. (2023, October). A machine learning approach to credit card customer segmentation for economic stability. In Proceedings of the 4th International Conference on Economic Management and Big Data Applications, ICEMBDA (pp. 27-29).

[8] Lee, M. S., Admoni, H., & Simmons, R. (2023). Closed-loop reasoning about counterfactuals to improve policy transparency. In International Conference o Machine Learning (ICML) Workshop on Counterfactuals in Minds and Machines.

[9] Zhu, W., Yao, Y., & Yang, J. (2025). Real-Time Risk Control Effects of Digital Compliance Dashboards: An Empirical Study Across Multiple Enterprises Using Process Mining, Anomaly Detection, and Interrupt Time Series.

[10] Karataiev, O., & Shubin, I. (2023). Formal model of multi-agent architecture of a software system based on knowledge interpretation. Radioelectronic and Computer Systems, (4), 53-64.

[11] Li, T., Xia, J., Liu, S., & Jiang, Y. (2025). Digital Transformation of Human Resources: From Consulting Frameworks to AI-Enabled Learning Management Systems.

[12] Age, P. (2024). Hybrid Methodologies for Studying Social and Cultural Memory in the. The Remaking of Memory in the Age of the Internet and Social Media, 241.

[13] Gu, X., Liu, M., & Yang, J. (2025). Application and Effectiveness Evaluation of Federated Learning Methods in Anti-Money Laundering Collaborative Modeling Across Inter-Institutional Transaction Networks.

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

[15] Shahriari, R., & Ragan, E. D. (2025). A Systematic Survey of Empirical User Studies of Unintentional Information Disclosure in Everyday Digital Interaction. arXiv preprint arXiv:2509.16003.

[16] Mao, Y., Ma, X., & Li, J. (2025). Research on Web System Anomaly Detection and Intelligent Operations Based on Log Modeling and Self-Supervised Learning.

[17] Gu, X., Yang, J., & Liu, M. (2025). Research on a Green Money Laundering Identification Framework and Risk Monitoring Mechanism Integrating Artificial Intelligence and Environmental Governance Data.

e-ISSN: 3079-6377 p-ISSN: 3079-6369

[18] Pásková, M., Štekerová, K., Zanker, M., Lasisi, T. T., & Zelenka, J. (2024). Water pollution generated by tourism: Review of system dynamics models. Heliyon, 10(1).

[19] Cai, B., Bai, W., Lu, Y., & Lu, K. (2024, June). Fuzz like a Pro: Using Auditor Knowledge to Detect Financial Vulnerabilities in Smart Contracts. In 2024 International Conference on Meta Computing (ICMC) (pp. 230-240). IEEE.

[20] Awadallah, A., Lara, Y., Magazine, R., Mozannar, H., Nambi, A., Pandya, Y., ... & Zhao, A. (2025). Fara-7B: An Efficient Agentic Model for Computer Use. arXiv preprint arXiv:2511.19663.

[21] 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.

[22] Sevetlidis, V., & Pavlidis, G. (2026). Training Memory in Deep Neural Networks: Mechanisms, Evidence, and Measurement Gaps. arXiv preprint arXiv:2601.21624.

[23] 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.

[24] Al-Bashrawi, M. A., Al-Sharafi, M. A., Elgendy, I. A., Helal, M. Y., Anbalagan, M. K., Chae, I., & Dwivedi, Y. K. (2026). Agentic AI systems and the future of entrepreneurship: a perspective on co-agency, innovation, and ecosystem transformation. International Entrepreneurship and Management Journal, 22(1), 27.

[25] Mao, Y., Ma, X., & Li, J. (2025). Research on API Security Gateway and Data Access Control Model for Multi-Tenant Full-Stack Systems.