LLM-MAD: Multi-Agent LLM Reasoning for Multi-Modal Shilling Attack Detection in Online Platforms

Dina Nawara[1] and Rasha Kashef [2]

Electrical, Computer and Biomedical Engineering Department
Toronto Metropolitan University, Toronto, Canada
{dina.nawara, rkashef}@torontomu.ca

Abstract. Shilling attackers often operate within or across social platforms, leveraging their influence to manipulate collective opinions, distort reputation metrics, and skew product visibility. Shilling attack detection in recommender systems remains a persistent challenge due to the increasing sophistication of adversarial profiles. Traditional detection pipelines, including statistical and supervised models, often fail to generalize across multimodal attacks that simultaneously target ratings, reviews, and user behaviour (i.e., both ratings and reviews). This paper presents LLM-MAD, a novel multi-agent detection framework that leverages prompt-driven Large Language Models (LLMs) for robust and generalizable detection of adversarial shilling attacks. Our framework, by dissecting behavioral signals across various modalities, models the implicit and explicit trust dynamics and the adversarial influence pathways that are central to social networks. Our proposed model deploys three specialized GPT-based agents, each focusing individually on reviews, ratings, and profile behaviour. The outputs from these agents are then fused using a meta-agent, which a rule-based orchestrator governs to issue final judgments and justifications. To ensure robustness across different adversarial strategies, we simulate three distinct attack scenarios (ratings-only, reviews-only, and profile-level) on constructed infected datasets, reflecting realistic obfuscation tactics. Evaluated on both the infected Amazon and Yelp datasets, LLM-MAD achieves high accuracy in tackling multi-modal shilling attacks. Experimental results demonstrate that our model outperforms both classical baselines and advanced hybrid models, achieving classification accuracies of 98% and 84.2% on the Amazon and Yelp datasets, respectively, while remaining cost-effective in deployment. Our work highlights the strength of LLM-driven agent collaboration in building resilient and transparent recommender systems under multi-modal adversarial conditions.

Keywords: Obfuscated Shilling Attack, Multi-Agents, LLMs.

1 Introduction

With the increasing scale of online platforms, recommender systems (RS) have become great tools for personalizing recommendations and optimizing user-item interactions [1]. These systems heavily rely on historical user-item feedback, specifically ratings and reviews, to infer preferences and enhance accuracy. As a result, recommender

systems are also attractive targets for adversarial manipulation. One common threat is the shilling attack, where fake user profiles are crafted and injected to manipulate the item's visibility and alter recommendation outcomes [2]. These attacks can involve boosting a product, which is called a push-attack, or deliberately demoting it, i.e., a nuke-attack, by assigning extreme ratings and simulating legitimate user behaviours. Traditional shilling generation methods rely on manually injected patterns with human intervention or rule-based heuristics, which leads to poor diversity and easy detection [3]. More recent research approaches employ data-driven models, such as Generative Adversarial Networks (GANs), which learn to generate rating patterns by mimicking genuine users [4], [5]. In addition to that, with advances in NLP and the integration of reviews into hybrid recommender systems, review manipulation poses a significant risk and requires focused development [6], [7]. Obfuscated shilling attacks leverage the capability to manipulate both the textual (i.e. reviews), and the numerical (ratings) dimensions of user feedback simultaneously. Unlike traditional attacks that target either ratings or reviews, these adversarial attacks coordinate across both modalities, aligning the sentiment, content, and rating score in a way that mimics genuine profiles, as illustrated in Figure 1. This multi-modal attack presents a significant detection challenge, as the textual and rating feedback reinforce each other, even misleading advanced detectors that are trained on uni-modal attacks. As these attacks became more consistent and human-like, they increased the vulnerability of existing detection pipelines. To address this challenge, this research introduces LLM-MAD, an LLM-driven multi-agent detection framework designed to defend against obfuscated and coordinated shilling attacks. Our approach begins by constructing infected datasets using two state-of-the-art generative pipelines from prior work (i) a GAN-based model for injecting realistic-looking ratings [4], and (ii) A BERT-GPT-based ensemble for review generation [6]. We craft the shilling profiles to tackle three scenarios (i.e., rating-only attack, review-only attack, and multi-modal attack), which represent real-world cases that reflect obfuscated shilling attacks. Once the infected datasets are constructed, detection is performed using our proposed model, which leverages the prompt-based and multi-agent capabilities of GPT-4. LLM-MAD incorporates three specialized GPT-based agents, where the first agent analyzes numerical patterns in user ratings, the second agent evaluates the linguistic authenticity of reviews, and the third agent assesses behavioural coherence at the profile level by analyzing both ratings and reviews. Each agent works independently using few-shot prompting to generate its prediction. The outputs of the three agents are then fused by a meta-agent, which consolidates signals across modalities in a unified output. Our proposed model incorporates a custom-designed orchestrator that interprets the fused output by leveraging GPT-4's prompt-based reasoning to generate the final verdict along with a natural language justification. To evaluate the effectiveness of our models in terms of accuracy and cost, we compared them against two approaches: (i) a hybrid detection model that combines BERT for text classification and XGBoost for ratings, and (ii) a variant of our system employing a different orchestration strategy using LangChain-based decision logic [8]. Our main contributions are:

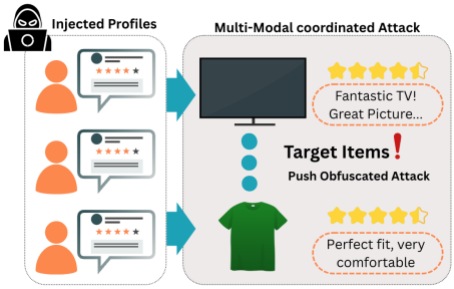

Fig. 1. Illustration of a multi-modal obfuscated shilling attack.

1. We introduce a multi-agent detection framework designed to identify obfuscated shilling attacks in recommender systems, where the model incorporates prompt-based GPT agents to independently analyze textual reviews, numerical ratings, and behavioural patterns in profiles.
2. We construct multimodal adversarial datasets that simulate three distinct attack scenarios: rating-only, review-only, and combined profile-level manipulation.
3. We develop a prompt-engineered orchestration mechanism in which a GPT-4 meta-agent consolidates the output of the specialized agents and produces a final classification label along with a textual justification.
4. We conduct an experimental evaluation against two baselines, (i) a BERT-XGBoost hybrid model and (ii) a LangChain-based orchestration variant of our architecture.

The rest of the paper is structured as follows: Section 2 surveys the related work, Section 3 explains the proposed model, Section 4 presents the experimental results, and Section 5 concludes with a discussion of future work.

2 Related Work

2.1 Shilling Attacks in Recommender Systems

Shilling attacks are well-studied security threats that severely impact the integrity of recommendation systems [9]. Where adversaries inject fabricated user profiles to manipulate item recommendations and rankings by either promoting the items, i.e., push attacks, or demoting them, i.e., nuke attacks. Shilling attack generation often relies on heuristics [10], for example, assigning extreme ratings to target items while populating filler items using random or average ratings. Some of the most commonly used shilling attacks are push attacks, which include bandwagon or segment-based attacks [11], or nuke shilling attacks, such as reverse bandwagon or love-hate attacks [12]. These attacks are simple to implement and deploy; however, they exhibit detectable patterns and limited behavioural diversity, making them less effective against robust defences. To increase undetectability, more advanced strategies are being explored; for example, recent work leverages the use of generative models [13], such

as Generative Adversarial Networks (GANs), to automatically generate rating vectors that closely mimic authentic user behaviour. These approaches train discriminators to refine profile realism by incorporating multi-objective loss functions, further increasing attack efficiency. For instance, in [14], the RWA-GAN system was proposed to introduce a neural shilling attack model that leverages Generative Adversarial Networks (GANs) to generate fake user profiles optimized for graph-based recommender systems. The authors designed a novel generator to minimize shilling loss and guide profile generation over the user-item interaction graph. In addition, in our prior work [4], we introduced a penalized GAN framework with latent perturbation, designed to generate realistic-looking rating vectors that better mimic authentic user behaviour. The generator was trained to introduce structured noise within the latent space, enabling the creation of obfuscated rating profiles that we build upon in this work.

Besides rating manipulation, a growing direction of research investigates text-based shilling attacks, where adversaries generate synthetic reviews to reinforce the intended manipulation. For instance, the researchers in [15] proposed a framework targeting Review-based Recommender systems. Their method employed a reinforcement learning agent to train a fake review generator that manipulates system outputs by inducing prediction shifts upon injection. Moreover, in earlier work [6], we introduced a dual-phase framework for generating and detecting synthetic reviews. The framework incorporated an ensemble of fine-tuned BERT and GPT-based text generation, establishing the foundation for simulating realistic review-based attacks, which we build upon in this work. Malicious attackers often combine generated reviews with rating vectors in multi-modal, coordinated attacks, where these obfuscated attacks increase the plausibility of fake profiles by synchronizing textual sentiment with numerical ratings, making detection even more challenging for well-trained detectors. While prior work has explored multi-modal attack generation or multi-feature detection independently, to the best of our knowledge, none have introduced a prompt-driven, modular detection framework that is capable of reasoning across modalities using multi-agent large language models (LLMs). Our proposed model, LLM-MAD, is the first to simulate and detect coordinated shilling attacks using a fully prompt-orchestrated pipeline, enabling explainable and adaptable defences that require minimal supervision.

2.2 Traditional Detection Methods

Early shilling detection methods mainly focused on identifying anomalies in rating patterns using statistical or heuristic-based filters. These approaches relied on features such as rating deviation, user profile similarity, filler size, and time-based patterns [16]. These methods were effective against simple attacks that exhibited clear behavioral signatures. To enhance robustness, a supervised learning approach was incorporated, training classifiers such as Support Vector Machines (SVMs), decision trees, and ensemble models using features extracted from the user-item extraction matrix [17]. For instance, in [18], the authors investigated the problem of detecting shilling attacks in recommender systems by employing One-Class Support Vector Machines (OCSVM)

as an anomaly detection approach. The authors used the MovieLens100K dataset for effectiveness analysis. It's known that supervised learning-based models achieve higher accuracy and more generalization across known attack types than statistical methods. However, they can't perform well against obfuscated attacks. Additionally, graph-based detectors have also been sought in the literature, leveraging the structural properties of user-item graphs to identify abnormal interactions [19]. For example, in [20], the authors addressed the vulnerabilities of GNN-based recommender systems, which are widely adopted, by proposing a two-stage training framework called Trust-GRS. This framework was designed to operate in a zero-knowledge setting to detect fake profiles. Their approach aimed to establish trustworthy aggregation and contrastive learning without relying on prior knowledge of the attack. Graph-based methods have limitations, as they focus on user-item connectivity and neighbourhood consistency in terms of ratings alone, without considering reviews, which limits their effectiveness in multi-modal coordinated attacks.

2.3 LLM-Based Agents in Recommender Systems

Large Language Models (LLMs) have shown remarkable capabilities in text generation and reasoning. They are now being embedded within agent frameworks. In these systems, an agent is defined as a modular, task-oriented unit that can perceive input, reason over it, and act or respond accordingly. For example, RecAgent [21] and Agent4Rec [22] proposed modular LLM-driven agents composed of profile, memory and decision components, allowing them to capture the user preferences and interaction dynamics in a more structured and personalized manner. The integration of LLM agents in recommender systems has reshaped traditional personalization methods [23], which relied on collaborative filtering, content-based filtering, or hybrid approaches. Instead of relying solely on historical interaction matrices, LLM agents can reason over text, user history and contextual features, making them suitable for more flexible and personalized recommendations [24]. In addition to simulation and personalization, LLM-based agents are being increasingly explored in the context of security threat generation and anomaly detection. Due to their powerful reasoning capabilities and their ability to handle unstructured multi-modal input. For example, in [25], the authors investigated the misuse of commercially available large language models (LLMs) for generating phishing attacks. They could generate online phishing emails using standard prompts. On the other hand, in [26], the authors proposed the ChatSpamDetector model, which leveraged agents to detect phishing emails through prompt-based analysis. This emerging direction suggests that LLM agents can serve as detectors of malicious intent, especially in domains where manipulation is implemented across modalities, such as in shilling attacks. Our work extends this by leveraging the agentic capabilities of LLMs to identify multi-modal obfuscation in recommender systems.

3 The Proposed Model

As traditional shilling attack detectors focus on single types, either textual or numerical, they fail to capture the holistic behaviour of malicious profiles that alter both data types. LLM-MAD is designed to address this limitation by breaking the detection process into modality-specific decisions and aggregating them through a reasoned consensus. Our detection framework is designed to identify multimodal shilling attacks that alter textual reviews and numerical ratings, as explained in Algorithm 1 and Figure 2. Let a user profile instance be represented as $x = (r, s, m)$, where r is the review text, $s \in R$ is the numeric rating, and m refers to the structured metadata. The classification task is to learn a function $f: \mathcal{X} \rightarrow \{\text{Real,Fake}\}$.

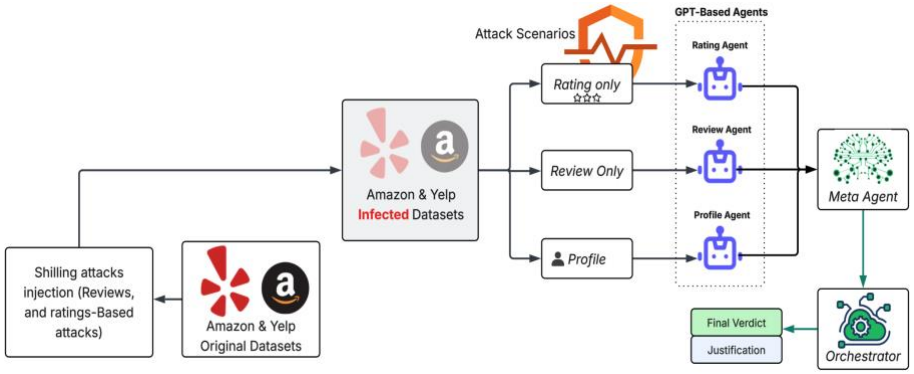

Fig. 2. Illustration of a multi-modal obfuscated shilling attack

Algorithm 1: LLM-MAD LLM-Based Multi-Agent Detector

Function: LLM-MAD_ORCHESTRATOR *(profile)*
Input: review_text ← *profile.reviewText*
rating ← *profile.rating*
metadata ← *{profile.source, profile.attack_type}*
Step 1: Agent Level Predictions
1. **Review_Agent:**
   - *review_output* ← *REVIEW_AGENT (review text)*
   return GPT 3.5 "Is this review human-written or AI-generated? Explain", *text*)
2. **Rating_Agent:**
   - *rating_output* ← *RATING_AGENT (rating)*
   return GPT 3.5 ("Is this rating suspicious or normal? Explain.", *value*)
3. **Profile_Agent:**
   - *rating_output* ← *RATING_AGENT (rating)*
   return GPT 3.5 ("Is this profile suspicious based on source and behavior? Explain." , *metadata*)

Step 2: Meta_Agent Reasoning
1. **Meta_Agent:**
   Input ← {review_output, rating_output, profile-output}
   return GPT 4 (" Given the justifications of the three agents, is the profile Real or Fake? Explain.", *agent _outputs*)
2. **Final Label Extraction** each output **do**
   a. *final_label* ← PARSE_LABEL
   b. *rationale* ← EXTRACT_JUSTIFICATION

Return
{final_label, rationale, review_output, rating_output, profile_output}

3.1 Modality Specific Agents

To support modular reasoning, we define three independent agents, each formed using GPT-3.5, and each responsible for evaluating a specific modality text.

- **Review Agent**: Receives the review text r and is prompted to assess whether the content is human-written or AI-generated.
- **Rating Agent:** Evaluates the numeric ratings in isolation, assessing whether they appear statistically or contextually anomalous.
- **Profile Agent**: Interprets metadata m to identify profile-level inconsistencies across ratings and textual reviews.

Each agent returns a binary label $\hat{y}_i \in$ {Real,Fake}and a textual explanation ρ_i presented in equation 1

$$A_i(x_i) \rightarrow (\hat{y}_i, \rho_i), \quad \text{for } i \in \{1,2,3\} \qquad (1)$$

3.2 Meta-Agent Aggregation

The agent outputs are passed to a GPT-4-based meta-agent, which synthesizes their justifications and issues the final verdict, where $\hat{y}$ is the final label and $\rho$is a multi-modal rationale explaining the classification. This step enables consensus even in the presence of partial disagreement:

$$M(\rho_1, \rho_2, \rho_3) \rightarrow (\hat{y}, \rho) \qquad (2)$$

We intentionally selected GPT-3.5 for the agents due to its speed and cost-effectiveness, and GPT-4 for the meta-agent due to its superior reasoning capabilities, contextual synthesis, and interpretability. This choice eliminates the need for supervised training or fine-tuning, and it enables flexible deployment across datasets and domains with the ability to generalize.

3.3 Orchestration Logic

The orchestration function executes the pipeline as shown in Algorithm 1, coordinating the dispatch of input, parsing agent responses, and performing meta-agent inference. The final output is a structured record of all agent predictions, meta-agent decisions, and associated rationales. Unlike static workflows such as LangChain, our custom manual orchestrator ensures control over agent prompts, structured output extraction, and consistent reasoning chains. We aimed to develop a customized orchestrator to reduce ambiguity in agent coordination and enable flexibility and effectiveness. The logic is implemented as a linear pipeline that coordinates input processing, agent interaction, and decision fusion. It begins by extracting the review text, numeric rating, and metadata from each profile instance. These components are then routed to their corresponding agents. Each agent returns a binary prediction along with a textual justification. The outputs are passed to the meta-agent, which synthesizes the justifications and produces a final verdict and rationale.

3.4 Evaluation

We evaluated LLM-MAD on two real-world datasets, namely Amazon Electronics [27] and Yelp [28], which were infected with multimodal shilling attacks involving GAN-based rating manipulation and LLM-generated reviews, as presented in our prior work. The detection task is to correctly classify each profile as real or fake, across three scenarios (i) rating-only attack, (ii) review-only attack, and (iii) coordinated profile attack. We compare LLM-MAD against two baselines: the first is the LangChain-Orchestrated variant of our proposed model, and the second is a classical hybrid (BERT + XGBoost) with a majority voting fusion. The model is benchmarked using accuracy, precision, recall, and F1score. In addition to that, agent-level interpretability metrics are incorporated:

- **Agent Agreement**, which is the proportion of instances where the three agents produce consistent local predictions.
- **The False Acceptance Rate (FAR)** is the percentage of fake profiles that are incorrectly classified as real.

4 Experimental Results

This section discusses the implementation setup, including datasets, agents, prompts, and main metrics.

4.1 Datasets

We evaluate the proposed LLM-MAD framework on subsets of two widely used real-world datasets: Amazon Electronics and Yelp. Both datasets contain structured user-item interactions, including numerical ratings and review text. To simulate realistic shilling attacks, we injected adversarial profiles using a hybrid generation pipeline from our prior work [4], [6]. For both Amazon and Yelp, we constructed injected datasets by adding 3,000 synthetic adversarial profiles to real user subsets, as described in Table 1.

Table 1. Datasets Statistics

Dataset	Real Subsets size	Infected Dataset size	Attack Size	Unique Users	Fake Profile	Real Profile
Amazon	24,272	27,273	11%	4,000	3,000	24,272
Yelp	20,077	23,077	13%	12,472	3,000	20,077

4.2 Experiment Configuration

All experiments were conducted using OpenAI's GPT-3.5 [29] and GPT-4 models via API access. The three modality-specific agents — i.e., review, rating, and profile — were formed using GPT-3.5, whereas the meta-agent responsible for aggregating agent justifications was deployed with GPT-4. For the LangChain baseline, all the agents and orchestration were managed using a chained GPT-4 pipeline to enable a direct architectural comparison. In the classical hybrid baseline approach, we use Microsoft pre-trained BERT from Hugging Face [30] to encode review features and XGBoost to

process structured data, with predictions aggregated via majority voting. For operating the agents, prompt templates are designed where they reflect the role and specialization of each agent in the detection pipeline. These templates enable agents to work in a task-specific or role-specific manner. This prompt-based formulation enables zero-shot and few-shot reasoning, facilitating adaptation across diverse domains. Figure 3 illustrates examples of the prompts used during the inference process.

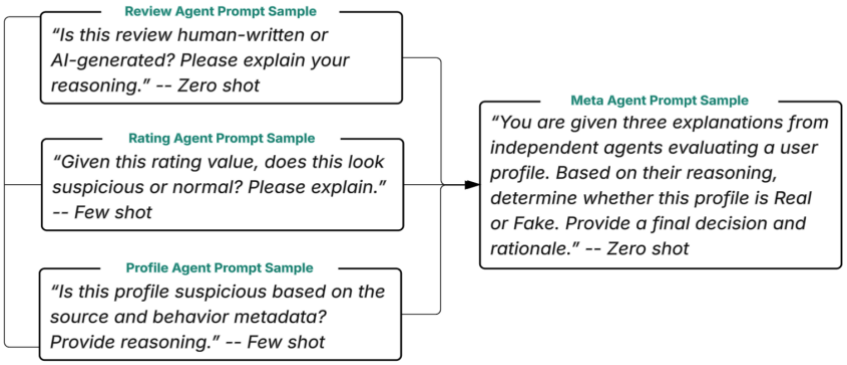

Fig.3. Example of the Used Task-Specific Prompt Templates

4.3 Experimental Results

We evaluated our proposed model against two baselines: (i) a LangChain-Orchestrated version, and (ii) a classical hybrid baseline combining BERT and XGBoost with majority voting. The evaluation metrics are divided into two categories: classification-level metrics, as shown in Tables 2 and Figure 5, and agent-level metrics for interpretability evaluation, presented in Tables 3 and 4.

For the **Amazon dataset** in Table 2 and Figure 5, LLM-MAD outperformed all baselines, achieving 98.00% accuracy, with a very high precision of 0.998 and a competitive F1 score of 0.797. Although the LangChain pipeline reported a higher F1 score of 0.867, it did so by sacrificing precision stability and requiring significantly more time per 1,000 instances, at 210 minutes compared to 45 minutes for LLM-MAD. The classical hybrid model achieved only 79.50% accuracy, with a weak recall of 0.2 and the lowest F1 score of 0.3, reflecting its limitations in capturing coordinated attack patterns. In Figure 4, the precision-recall curve indicates that LLM-MAD consistently maintains higher precision across varying recall levels.

For the **Yelp dataset**, LLM-MAD again achieved the highest accuracy of 84.20% and a strong balance between precision (0.90) and recall (0.842), resulting in an F1 score of 0.853. In contrast, the LangChain-Orchestrated pipeline exhibited a major degradation, achieving only 20.00% accuracy with unbalanced precision and recall values. This highlights the instability of chained prompting when applied to noisy or sparse domains.

As seen in Tables 3 and 4, LLM-MAD consistently delivered the most interpretable and stable behaviour across agents. On **the Amazon dataset**, the agent agreement reached 95.91%, and the accuracy of the review and rating agents exceeded 93%. The full attack detection accuracy was 98.03%, supported by a strong ROC-AUC of 0.831 and a false acceptance rate of 0.00%. In contrast, the LangChain pipeline exhibited a lower agreement of 70.70% and a higher false alarm rate (FAR) of 0.10%. The classical hybrid model failed to detect coordinated attacks (yielding 0% accuracy on profile attacks) and lacked an explainability mechanism beyond the prediction output. For the **Yelp dataset**, the agent agreement in LLM-MAD dropped to 54.00%, reflecting the noisier nature of the Yelp data and its more diverse linguistic and behavioural patterns. Despite that, LLM-MAD maintained a high ROC-AUC of 0.84 and outperformed the other two models.

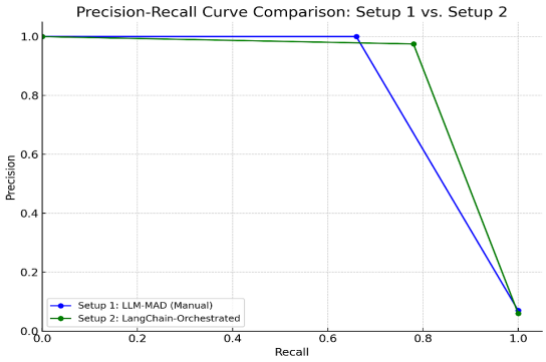

Fig. 4. Precision-recall curve comparison between LLM-MAD & LangChain-Orchestrated

Table 2. Classification Metrics- Accuracy

Dataset	Detector Name	Orchestrator	Agents Used	Meta-Agent	Accuracy
Amazon	**LLM-MAD**	Custom	GPT-3.5/GPT-4	GPT-4	**98.00%**
	LangChain- Orch	LangChain	GPT-4	GPT-4	76.60%
	Hybrid (BERT+XGB)	None	BERT + XGBoost	Majority Voting	79.50%
Yelp	**LLM-MAD**	Custom	GPT-3.5/GPT-4	GPT-4	**84.20%**
	LangChain- Orch	LangChain	GPT-4	GPT-4	20.00%
	Hybrid (BERT+XGB)	None	BERT + XGBoost	Majority Voting	76.90%

Table 3. Agent Level and Interpretability Metrics

Dataset	Detector Name	Review Agent Acc	Rating Agent Acc	Full Attack Agent Acc
Amazon	**LLM-MAD**	**93.95%**	**94.18%**	98.03%
	LangChain-Orchestrated	92.11%	94.12%	99.99%
	Hybrid (BERT+XGB)	93%	92.38%	0% (no profile agent)
Yelp	**LLM-MAD**	**76.60%**	**70.70%**	**97.20%**
	LangChain-Orchestrated	50.00%	50.10%	53.00%
	Hybrid (BERT+XGB)	76%	75.90%	0% (no profile agent)

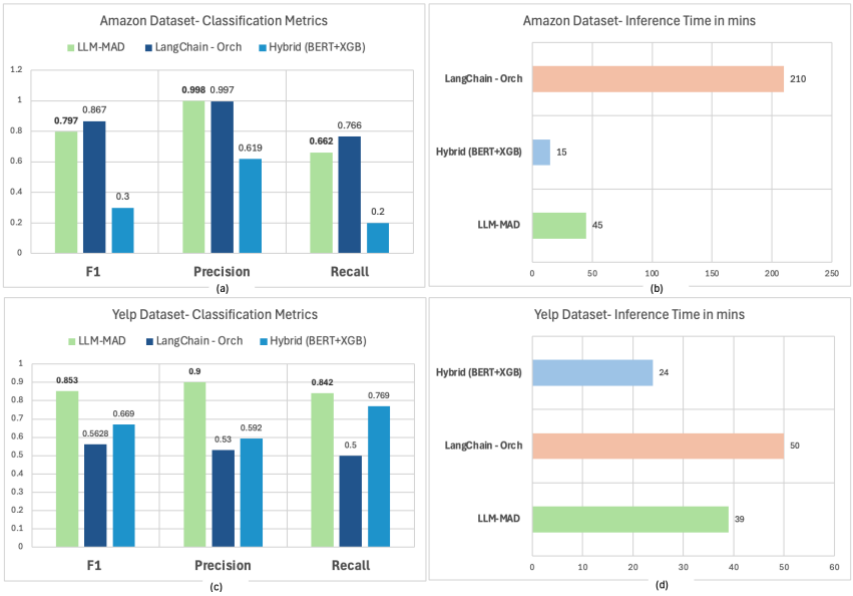

Fig. 5. Overall Classification Metrics (a)(b) Amazon Dataset and (c)(d) Yelp Dataset

We also evaluated the model in terms of cost, as described in Table 5, to be a reference for researchers and readers. Despite using GPT-4 for meta-reasoning, LLM-MAD was significantly more efficient than LangChain, with total inference time per 1,000 profiles reduced by over 75%. Token analysis revealed that LLM-MAD processed ~750,000 tokens for $12.18, compared to a similar or higher cost for the LangChain variant, which operated less efficiently at around 550,000 tokens but required more expensive GPT-4 calls throughout the pipeline.

Table 4. Agent-level Agreement, ROC-AUC and FAR Metrics

Dataset	Detector Name	Agent Agreement	ROC-AUC	FAR
Amazon	**LLM-MAD**	**95.91%**	**0.831**	**0.00%**
	LangChain-Orchestrated	70.70%	0.82	0.10%
	Hybrid (BERT+XGB)	N/A	0.62	>3%
Yelp	**LLM-MAD**	**54.00%**	**0.84**	0.20%
	LangChain-Orchestrated	11.90%	0.512	>3%
	Hybrid (BERT+XGB)	N/A	0.5	>3%

Table 5. Cost Analysis per Experimental Setup

Setup	Model Configuration	Total Tokens	Cost (USD)
LLM-MAD	(GPT-3.5 Agent+ GPT-4 meta)	750,000	$12.18
LangChain-Orchestrated	All GPT-4 LangChain	~550,000	$12.18+
Hybrid (BERT+XGB)	BERT + XGBoost (baseline)	N/A	$0.00 (open source)

5 Conclusion and Future Work

This paper presents LLM-MAD, a modular, prompt-driven framework for detecting multimodal shilling attacks in recommender systems. Traditional detection methods often fail to identify coordinated shilling attacks and adversarial behaviour that simultaneously target both textual reviews and numerical ratings. To address this, we proposed a multi-agent architecture composed of specialized GPT-3.5-based agents, where each is dedicated to a specific modality, and a GPT-4 serves as a meta-agent responsible for fusing justifications and issuing a final verdict. A custom orchestration layer is designed to govern this interaction. We evaluated LLM-MAD across two real-world domains, Amazon Electronics and Yelp, using datasets infected with adversarial profiles generated through GAN-based rating perturbations and LLM-generated reviews. The attacks are generated over real-world user and item data that introduce obfuscation and semantic consistency. This allows for controlled benchmarking under complex threat secnarios. The proposed framework demonstrated strong generalization, achieving high classification accuracy, precision, and a stable F1 performance across multiple attack scenarios. Compared to both a LangChain-Orchestrated variant and a classical hybrid baseline (BERT + XGBoost), LLM-MAD achieved superior detection metrics while maintaining cost efficiency and interpretability. While LLM-MAD leverages closed-source APIs such as GPT-3.5 and GPT-4, this design choice enables accurate contextual reasoning and interpretability with zero-shot flexibility across different domains. In future work, we aim to explore reinforcement-based refinement strategies that enable the agents to iteratively improve their decision-making based on feedback and context. In addition to that, we intend to explore open-source alternatives such as LLaMA-3 or Mistral.

References

1. A. Shankar, P. Perumal, M. Subramanian, N. Ramu, D. Natesan, V. R. Kulkarni and T. Stephan, "An intelligent recommendation system in e-commerce using ensemble learning," *Multimedia Tools and Applications,* vol. 83, p. 48521–48537, 2023.
2. D. Nawara, A. Aly and R. Kashef, "Shilling Attacks and Fake Reviews Injection: Principles, Models, and Datasets," *IEEE Transactions on Computational Social Systems,* vol. 12, no. 1, 2024.
3. C. Huang and H. Li, "Single-User Injection for Invisible Shilling Attack against Recommender Systems," in *CIKM '23: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* , 2023.
4. D. Nawara and R. Kashef, "Penalized GANs with Latent Perturbation for Robust Shilling Attack Generation in Recommender Systems," *Preprint-Research Square,* no. 13, 2025.
5. Z. Wang, M. Gao, J. Li, J. Zhang and J. Zhong, "Gray-Box Shilling Attack: An Adversarial Learning Approach," *ACM Transactions on Intelligent Systems and Technology,* vol. 13, no. 5, 2022.
6. D. Nawara and R. Kashef, "A dual-phase framework for detecting authentic and computer-generated customer reviews using large language models," *Decision Analytics Journal,* vol. 15, 2025.

7. J. Salminen, C. Kandpal, A. M. Kamel, S.-g. Jung and B. J. Jansen, "Creating and detecting fake reviews of online products," *Journal of Retailing and Consumer Services,* vol. 64, 2022.
8. H. Chase, "LangChain," 2022. [Online]. Available: https://github.com/hwchase17/langchain. [Accessed 2025].
9. S. Guo, T. Bai and W. Deng, "Targeted Shilling Attacks on GNN-based Recommender Systems," in *CIKM '23: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* , 2023.
10. F. Rezaimehr and C. Dadkhah, "Injection Shilling Attack Tool for Recommender Systems," in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, 2021.
11. S. Bansal and N. Baliyan, "A Multi-criteria Evaluation of Evolutionary Algorithms Against Segment Based Shilling Attacks," in *Advances in Intelligent Systems and Computing*, 2021.
12. P. Narayanan and V. K., "Hybrid CNN and RNN-based shilling attack framework in social recommender networks," *EAI Endorsed Transactions on Scalable Information Systems,* vol. 9, no. 35, 2022.
13. J. Barbieri, L. G. Alvim, F. Braida and G. Zimbrão, "Simulating real profiles for shilling attacks: A generative approach," *Knowledge-Based Systems,* vol. 230, 3032.
14. S. Liu, S. Yu, H. Li, Z. Yang, M. Duan and X. Liao, "A novel shilling attack on black-box recommendation systems for multiple targets," *Neural Computing and Applications ,* vol. 37, 2024.
15. H.-Y. Chiang, Y.-S. Chen, Y.-Z. Song, H.-H. Shuai and J. S. Chang, "Shilling Black-box Review-based Recommender Systems through Fake Review Generation," in *KDD '23: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* , 2023.
16. R. A. Zayed, L. F. Ibrahim, H. A. Hefny, H. A. Salman and A. AlMohimeed, "Experimental and Theoretical Study for the Popular Shilling Attacks Detection Methods in Collaborative Recommender System," *IEEE Access,* vol. 11, 2023.
17. P. K. Singh, P. K. D. Pramanik, N. Sinhababu and P. Choudhury, "Detecting Unknown Shilling Attacks in Recommendation Systems," *Wireless Personal Communications ,* vol. 137, 2024.
18. H. İ. Ayaz and Z. K. Öztürk, "Shilling Attack Detection with One Class Support Vector Machines," *NEU Fen Muh Bil Der,* vol. 5, no. 2, 2023.
19. Y. Zhang, Q. Hao, W. Zheng and Y. Xiao, "User similarity-based graph convolutional neural network for shilling attack detection," *Applied Intelligence ,* vol. 55, 2025.
20. L. Mu, Z. Liu, Z. Zhu and Z. Lin, "Trust-GRS: A Trustworthy Training Framework for Graph Neural Network Based Recommender Systems Against Shilling Attacks," in *AAAI Technical Track on Data Mining & Knowledge Management II*, 2025.

21. L. Wang, J. Zhang, H. Yang, Z. Chen, J. Tang, Z. Zhang, X. Chen, Y. Lin, R. Song, W. X. Zhao, J. Xu, Z. Dou, J. Wang and J.-R. Wen, *User Behavior Simulation with Large Language Model based Agents,* ArXiv:CS preprint, 2024.
22. L. Wang, J. Zhang, H. Yang, Z.-Y. Chen, J. Tang, Z. Zhang, X. Chen, Y. Lin, H. Sun, R. Song, X. Zhao, J. Xu, Z. Dou, J. Wang and J.-R. Wen, "User Behavior Simulation with Large Language Model-based Agent," *ACM Transactions on Information Systems,* vol. 43, no. 2, 2025.
23. S. Xu, W. Hua and Y. Zhang, "OpenP5: An Open-Source Platform for Developing, Training, and Evaluating LLM-based Recommender Systems," in *SIGIR '24: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.
24. N. Dehbozorgi, M. T. Kunuku and S. Pouriyeh, "Personalized Pedagogy Through a LLM-Based Recommender System," in *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky* , 2024.
25. S. S. Roy, P. Thota, K. V. Naragam and S. Nilizadeh, "From Chatbots to Phishbots?: Phishing Scam Generation in Commercial Large Language Models," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024.
26. N. F. H. N. D. C. Takashi Koide, *ChatSpamDetector: Leveraging Large Language Models for Effective Phishing Email Detection,* arXiv: cs preprint, 2024.
27. "Amazon Dataset," [Online]. Available: https://nijianmo.github.io/amazon/index.html. [Accessed 2025].
28. "Yelp Dataset," [Online]. Available: https://www.kaggle.com/datasets/yelp-dataset/yelp- dataset. [Accessed 2025].
29. "OpenAi," [Online]. Available: https://platform.openai.com/docs/models/gpt-3.5-turbo.
30. "deberta," [Online]. Available: https://huggingface.co/microsoft/deberta-v3-small. [Accessed 2025].