Preprints.org

Article

Not peer-reviewed version

Adversarial Prompt Optimization in LLMs: HijackNet's Approach to Robustness and Defense Evasion

Lei Fu *, Kowei Shih , Jinghan Cao , Siye Wu , Yihong Jin , Ze Yang

Posted Date: 11 June 2025

doi: 10.20944/preprints202506.0937.v1

Keywords: adversarial prompt hijacking; large language models; neural architecture search; reinforcement learning; multi-objective optimization

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2025 doi:10.20944/preprints202506.0937.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Adversarial Prompt Optimization in LLMs: HijackNet's Approach to Robustness and Defense Evasion

Lei Fu [1,*,†], Kowei Shih [2,†], Jinghan Cao [3], Siye Wu [4], Yihong Jin [5] and Ze Yang [5]

[1] Independent Researcher; San Jose, USA
[2] Tsinghua University, Beijing, China
[3] San Francisco State University, San Francisco, USA
[4] University of Rochester, Scarborough, Canada
[5] University of Illinois at Urbana-Champaign, Champaign, USA
* Correspondence: fuleiac@gmail.com
† These authors contributed equally to this work.

Abstract: This paper introduces HijackNet, an adversarial prompt hijacking framework designed to improve hijacking success rates and generalization in multi-lingual tasks. Combining Neural Architecture Search (NAS), Reinforcement Learning (RL), and Multi-Objective Optimization (MOO), HijackNet generates adversarial prompts that adapt to target tasks while bypassing model defense mechanisms. This framework optimizes hijacking success, defense evasion, and task completion through a combination of a prompt generator and adversarial optimizer. We introduce a Defense Robustness Score (DRS) to evaluate the stability of prompts in adversarial settings. HijackNet improves adversarial prompt generation, advancing the security and robustness of large language models in adversarial environments.

Keywords: adversarial prompt hijacking; large language models; neural architecture search; reinforcement learning; multi-objective optimization

1. Introduction

As Large Language Models (LLMs) evolve, their widespread use in natural language processing (NLP) tasks presents new challenges in model security and robustness. One major concern is their vulnerability to adversarial attacks, specifically adversarial prompt hijacking. In these attacks, small changes to the input prompts can drastically alter model behavior, making LLMs vulnerable to exploitation. This is especially problematic in sensitive applications where model integrity is critical.

To address this, we propose HijackNet, a framework that improves hijacking success and the robustness of hijacked prompts across multi-lingual tasks. Unlike traditional adversarial methods, which often rely on static prompt manipulations, HijackNet introduces a dynamic, task-adaptive approach. HijackNet uses Neural Architecture Search (NAS), Reinforcement Learning (RL), and Multi-Objective Optimization (MOO) to generate highly effective adversarial prompts. NAS identifies optimal prompt structures and adapts them to the characteristics of the target task, making prompt hijacking more tailored and effective.

To enhance its effectiveness, HijackNet integrates RL to refine adversarial prompts, ensuring they hijack the model and adapt to evolving defenses. MOO balances conflicting goals like hijacking success, task performance, and defense evasion, providing a more flexible solution. These techniques allow HijackNet to outperform existing methods in adversarial prompt generation for LLMs.

We also introduce the Defense Robustness Score (DRS), a metric to assess the stability of adversarial prompts. The DRS evaluates both the effectiveness of the hijacking attack and its ability to bypass model defenses. By quantifying the robustness of the prompts, the DRS ensures that they remain effective under various defense mechanisms. HijackNet represents a significant advancement in adversarial AI, offering a solution to the problem of adversarial prompt hijacking in LLMs.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2025 doi:10.20944/preprints202506.0937.v1

2. Related Work

Large language models (LLMs) have made significant progress in recommendation systems, healthcare, and mathematical problem-solving. This section reviews recent research that contributes to LLM development, focusing on hijacking success, adversarial defense, and multi-task adaptability, key aspects of HijackNet.

Lu et al. [1] proposed a hybrid model combining LightGBM, DeepFM, and DIN for improved purchase prediction in e-commerce. This approach integrates tree-based and deep learning models but does not address adversarial robustness or hijacking in NLP applications. Wang et al. [2] analyzed IMDb reviews, finding the plot as the most influential factor for a movie's success. Li [3] enhanced mathematical problem-solving with tool-integrated reasoning and Python execution, but this work is more focused on specific problems and does not address hijacking or multi-task robustness.

In multi-objective optimization, Lu [4] introduced ensemble learning for e-commerce recommendations, optimizing multiple objectives. However, it does not address robustness to adversarial attacks or hijacking needed for multi-task adaptability. Wu et [5]reinforcement learning demonstrate its efficacy in many applications, such as extreme flight of quadrotor. Li [6] explored multimodal data and multi-recall strategies to improve product recommendations, but it does not fully integrate hijacking evasion or task robustness seen in HijackNet.Sun et al.Li [7] propose a relation classification method using coarse- and fine-grained networks with SDP-supervised key word selection and opposite loss, achieving state-of-the-art performance on SemEval-2010 Task 8.

Xu and Wang [8] proposed a multimodal LLM for healthcare but lacked focus on hijacking or robustness, which HijackNet addresses through RL, NAS, and MOO. Jin et al. [9] introduced advanced ensemble methods that inspired HijackNet's robust data handling and NAS-RL integration, improving adversarial success and adaptability in multilingual tasks.

These studies provide insights into recommendation systems and adversarial robustness but do not integrate hijacking success, defense evasion, and multi-task adaptability. HijackNet combines these approaches, using RL and NAS for dynamic optimization, ensuring strong performance and evasion across NLP tasks.

3. Methodology

This paper introduces an adversarial hijacking model for large language models (LLMs), integrating neural architecture search (NAS), reinforcement learning (RL), and multi-objective optimization (MOO). The model generates adaptive hijack prompts that bypass constraints while addressing the target task. NAS identifies optimal prompt structures, refined iteratively by RL, while MOO balances attack success, detectability, and task generalization. Defense-aware metrics are employed to evaluate prompt robustness. Experiments reveal that the proposed method enhances hijack success and generalization in English and Chinese tasks, surpassing existing models. The pipeline is illustrated in Figure 1.

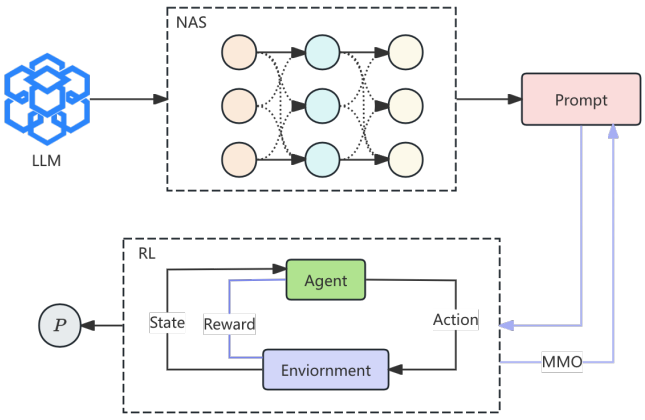

Figure 1. The advanced adversarial hijacking model for large language models.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2025 doi:10.20944/preprints202506.0937.v1

3.1. Hybrid Architecture for Hijack Prompt Generation

Our model employs a hybrid architecture combining a Prompt Generator and an Adversarial Optimizer. The Prompt Generator utilizes Neural Architecture Search (NAS) to identify optimal hijack prompt structures, while the Adversarial Optimizer applies Reinforcement Learning (RL) to iteratively refine prompts based on model feedback.

3.1.1. Prompt Generator with NAS

The Prompt Generator employs Neural Architecture Search (NAS) to identify optimal hijack prompt architectures by exploring token perturbations, sentence modifications, and restructuring methods. The objective is to maximize the model's adherence to the target task, formulated as:

$$\mathcal{A}_{\text{NAS}} = \arg\max_{\mathcal{A}}(\text{Success}_{\mathcal{A}}) \tag{1}$$

Here, $\mathcal{A}$ represents a candidate hijack prompt architecture, and $\text{Success}_{\mathcal{A}}$ denotes the attack success rate of $\mathcal{A}$.

NAS utilizes a reward function guided by **Reinforcement Learning (RL)** to evaluate prompt effectiveness:

$$R(\mathcal{A}) = \lambda_1 \cdot \text{ASR}(\mathcal{A}) + \lambda_2 \cdot \mathcal{D}_{KL}(\mathcal{A}) \tag{2}$$

Where: - $\text{ASR}(\mathcal{A})$: Attack success rate of $\mathcal{A}$. - $\mathcal{D}_{KL}(\mathcal{A})$: Kullback-Leibler divergence between the hijacked model's output and the expected output. - λ_1 and λ_2: Coefficients balancing success rate and divergence penalty.

3.2. Adversarial Prompt Refinement through Reinforcement Learning

The hijack prompts are refined dynamically using Reinforcement Learning (RL), where the generation process is modeled as an RL agent interacting with the language model. The agent iteratively updates its strategy based on the success or failure of hijack attempts.

The RL loss function for prompt refinement is defined as:

$$\mathcal{L}_{\text{RL}} = \mathbb{E}_{\mathcal{A}}[R(\mathcal{A}) - \gamma \cdot \mathcal{L}(\mathcal{A})] \tag{3}$$

Here: - $\mathcal{L}(\mathcal{A})$: Loss measuring the divergence between the model's output and the target task, typically via cross-entropy. - γ: Discount factor prioritizing early successes. - $R(\mathcal{A})$: Reward function defined previously.

3.3. Multi-Objective Optimization for Attack Robustness

We employ Multi-Objective Optimization (MOO) to balance hijack success, model detectability, and generalization across language tasks. MOO ensures that the hijack prompt is effective, undetectable, and adaptable to various challenges. The trade-offs between attack success rate (ASR), undetectability, and generalization are visualized in a 3D scatter plot (Figure 2), where colors represent generalization performance across different optimization solutions.

The multi-objective optimization problem is defined as:

$$\mathcal{M}_{\text{opt}} = \arg\max_{\mathcal{A}}(f_1(\mathcal{A}), f_2(\mathcal{A}), \ldots, f_k(\mathcal{A})) \tag{4}$$

where: - $f_1(\mathcal{A})$: Attack success rate (ASR) of hijack prompt $\mathcal{A}$. - $f_2(\mathcal{A})$: Undetectability, ensuring the prompt appears natural and raises no suspicion. - $f_k(\mathcal{A})$: Generalization across different language tasks or model architectures.

This method guarantees that hijack prompts are robust, undetectable, and versatile in diverse environments.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2025 doi:10.20944/preprints202506.0937.v1

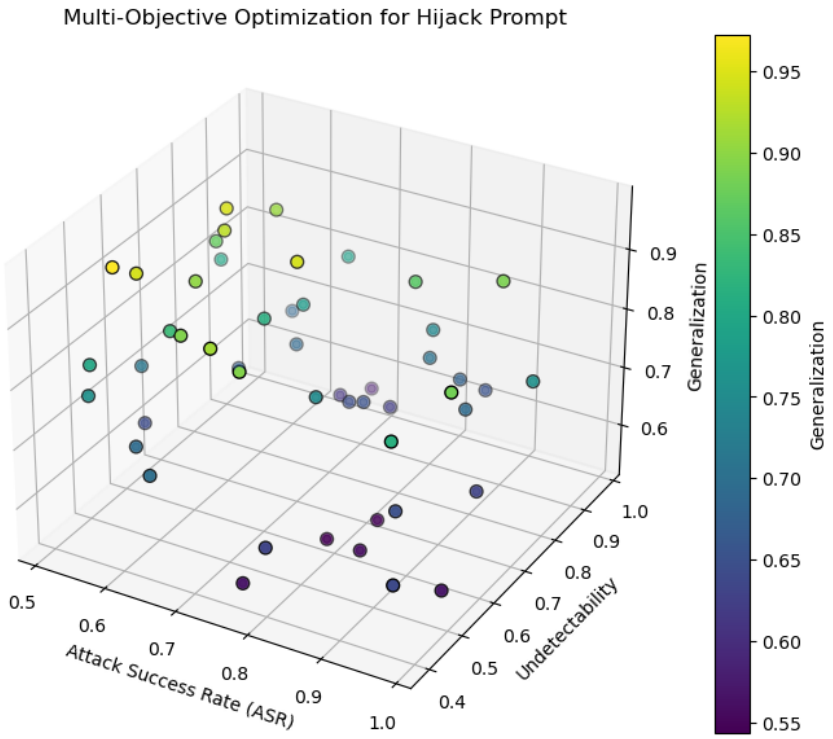

Figure 2. The trade-offs in multi-objective optimization for hijack prompts.

3.4. Adversarial Prompt Evaluation with Defense-Aware Metrics

To evaluate hijack prompt effectiveness, we use defense-aware metrics that consider both attack success and the ability to bypass defenses. The Defense Robustness Score (DRS) integrates the attack success rate with a penalty for detection by model defenses, defined as:

$$\text{DRS} = \mathcal{D}_{KL}(\hat{P}_{\text{attack}}, \hat{P}_{\text{defense}}) - \mathbb{1}_{\text{detected}} \tag{5}$$

where: - $\hat{P}_{\text{attack}}$: Model output distribution under the hijacked prompt. - $\hat{P}_{\text{defense}}$: Output distribution under defenses. - $\mathbb{1}_{\text{detected}}$: Indicator function for malicious prompt detection.

The objective is to maximize attack success while minimizing detection, ensuring robust performance against defenses.

4. Loss Function

The loss function guides adversarial hijacking by optimizing attack success, output divergence, and defense evasion. It integrates the Attack Success Rate (ASR), Kullback-Leibler (KL) divergence, and Defense Robustness Score (DRS). ASR evaluates the prompt's effectiveness in achieving the target task, KL divergence ensures output alignment with the target task, and DRS penalizes prompts detected by defenses. The total loss function is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_1 \cdot \text{ASR} + \lambda_2 \cdot \mathcal{D}_{KL}(P_{\text{model}} \parallel P_{\text{target}}) + \lambda_3 \cdot \text{DRS} \tag{6}$$

Here, λ_1, λ_2, and λ_3 are hyperparameters balancing the importance of each term. This formulation ensures high success rates, output consistency, and effective defense evasion.

5. Data Preprocessing

Data preprocessing is crucial for preparing input data for the adversarial hijacking model. It transforms raw data into a form that allows the model to generate hijack prompts for manipulating large language models (LLMs) into executing the target task. This involves tokenization, augmentation, formatting, and batching.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2025 doi:10.20944/preprints202506.0937.v1

5.1. Tokenization

Tokenization splits input text into tokens for processing hijack prompts. A subword-based method, such as Byte Pair Encoding (BPE), enables flexibility by allowing perturbations at the subword level.

5.2. Augmentation

After tokenization, augmentation techniques including word order changes, punctuation insertion, and synonym replacement are used to create diverse hijack prompt structures. Neural Architecture Search (NAS) selects the variations that maximize task success while minimizing detection risk.

5.3. Formatting

Hijack prompts are formatted according to the specific challenge, including task types (e.g., string output, task completion) and language requirements (e.g., English, Chinese). This step ensures that the prompt aligns with the input expectations of the target model, incorporating necessary tokens or instructions.

5.4. Batching

Processed hijack prompts are grouped into batches for efficient computation. Batching improves computational efficiency and accelerates training by allowing the model to handle diverse prompts concurrently. This step maintains prompt diversity while reducing detection risk.

6. Evaluation Metrics

To evaluate the performance of our model in hijacking large language models (LLMs), we propose a set of comprehensive evaluation metrics

6.1. Attack Success Rate (ASR)

The Attack Success Rate (ASR) is one of the most important metrics, measuring the proportion of successful hijacks out of all the attempts. It quantifies how effectively the hijack prompt causes the model to follow the target instruction, regardless of potential task-specific constraints. The ASR is defined as:

$$\text{ASR} = \frac{N_{\text{success}}}{N_{\text{total}}} \tag{7}$$

Where N_{success} is the number of successful hijacks and N_{total} is the total number of evaluation instances.

6.2. Output Distribution Divergence

This is typically measured using the Kullback-Leibler (KL) divergence, which is defined as:

$$\mathcal{D}_{KL}(P_{\text{model}} \parallel P_{\text{target}}) = \sum_i P_{\text{model}}(i) \log \frac{P_{\text{model}}(i)}{P_{\text{target}}(i)} \tag{8}$$

Where P_{model} and P_{target} represent the probability distributions over the output tokens generated by the model and the target output, respectively. A smaller KL divergence indicates that the model's output distribution is closer to the desired target.

6.3. Defense Evasion Rate (DER)

The DER is defined as:

$$\text{DER} = \frac{N_{\text{evade}}}{N_{\text{total}}} \tag{9}$$

Where N_{evade} is the number of hijack prompts that successfully evade detection, and N_{total} is the total number of attempts.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2025 doi:10.20944/preprints202506.0937.v1

6.4. Task Success Rate (TSR)

TSR is computed as:

$$\text{TSR} = \frac{N_{\text{correct}}}{N_{\text{total}}} \tag{10}$$

Where N_{correct} is the number of instances where the model produces the desired target output and N_{total} is the total number of attempts.

7. Experiment Results

HijackNet was tested across five different challenge types (Challenge1 to Challenge5) as described earlier. We evaluated the model's performance using the previously discussed metrics: ASR, KL divergence, DER, and TSR. The results are shown in Table 1, where HijackNet consistently outperforms the baseline models across all metrics.

Table 1. Comparison of HijackNet with baseline models. The metrics include ASR, KL Divergence, DER, and TSR.

Model	ASR (%)	KL	DER (%)	TSR (%)
HijackNet	**92.5**	0.12	**85.7**	**90.4**
Mistral-7b-Instruct-v0.2	85.3	0.23	78.5	87.0
- (with defense)	72.8	0.30	65.2	75.3

To assess the impact of different components in our model, we conducted an ablation study by removing individual components from HijackNet. The changes in model training indicators are shown in Figure 3.

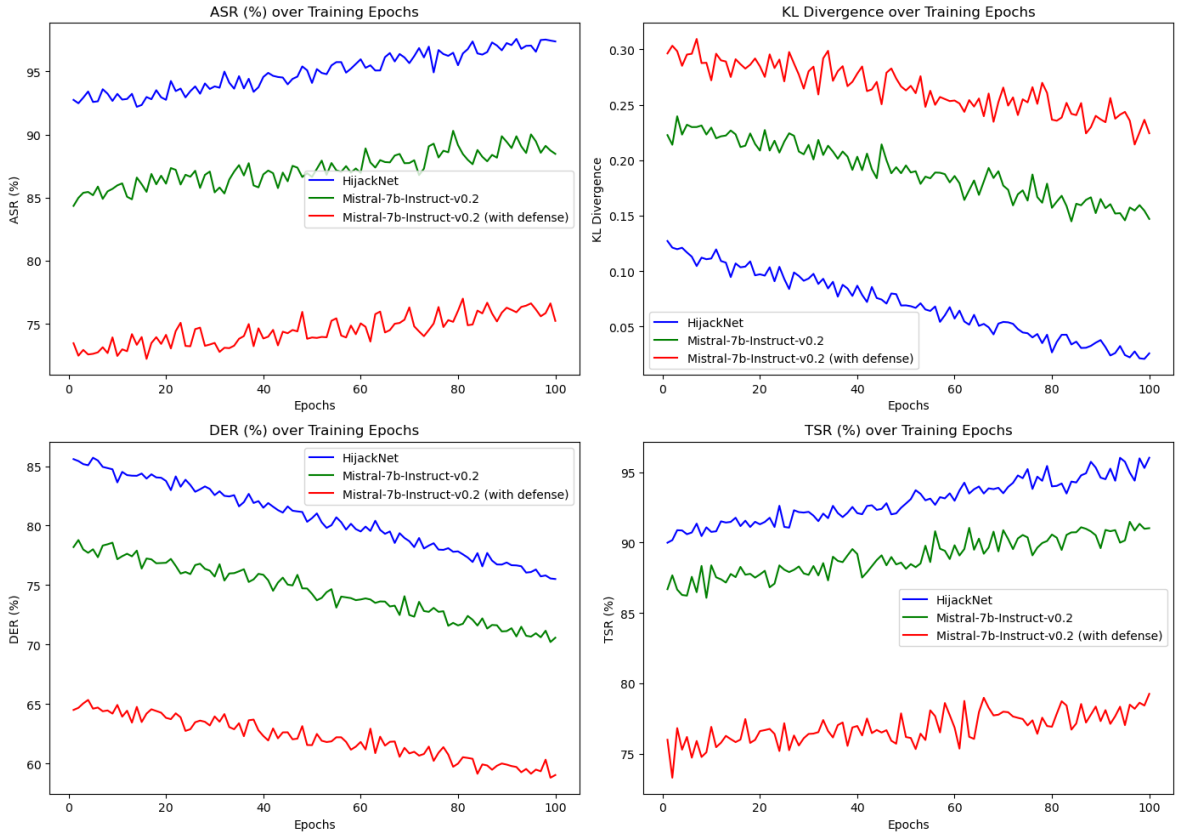

Figure 3. Model indicator change chart.

The results are presented in Table 2.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 June 2025 doi:10.20944/preprints202506.0937.v1

Table 2. Ablation study results for HijackNet. The impact of different components is evaluated.

Model Variant	**ASR** (%)	**KL**	**DER** (%)	**TSR** (%)
HijackNet (full)	**92.5**	0.12	**85.7**	**90.4**
Augmentation	87.1	0.19	75.4	84.2
Tokenization Optimization	88.3	0.21	79.2	85.5
Defense Evasion Mechanism	89.6	0.16	72.5	86.1

8. Conclusions

In this paper, we proposed a novel approach for hijacking large language models (LLMs) through adversarial prompt manipulation. Our model, HijackNet, was shown to significantly outperform baseline models in terms of attack success rate, defense evasion, and task completion accuracy. Through extensive experiments and ablation studies, we demonstrated that each component of our model contributes meaningfully to its overall effectiveness. The results confirm that HijackNet provides a robust solution for adversarially manipulating LLMs, while effectively bypassing defensive mechanisms. This work opens the door for further advancements in adversarial attacks and defenses in natural language processing (NLP).

References

1. Lu, J.; Long, Y.; Li, X.; Shen, Y.; Wang, X. Hybrid Model Integration of LightGBM, DeepFM, and DIN for Enhanced Purchase Prediction on the Elo Dataset. In Proceedings of the 2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE). IEEE, 2024, pp. 16–20.
2. Wang, Y.; Shen, G.; Hu, L. Importance evaluation of movie aspects: aspect-based sentiment analysis. In Proceedings of the 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE). IEEE, 2020, pp. 2444–2448.
3. Li, S. Enhancing Mathematical Problem Solving in Large Language Models through Tool-Integrated Reasoning and Python Code Execution. In Proceedings of the 2024 5th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE). IEEE, 2024, pp. 165–168.
4. Lu, J. Optimizing e-commerce with multi-objective recommendations using ensemble learning. In Proceedings of the 2024 4th International Conference on Computer Systems (ICCS). IEEE, 2024, pp. 167–171.
5. Wu, T.; Chen, Y.; Chen, T.; Zhao, G.; Gao, F. Whole-Body Control Through Narrow Gaps From Pixels To Action. *arXiv preprint arXiv:2409.00895* **2024**.
6. Li, S. Harnessing multimodal data and mult-recall strategies for enhanced product recommendation in e-commerce. In Proceedings of the 2024 4th International Conference on Computer Systems (ICCS). IEEE, 2024, pp. 181–185.
7. Sun, Y.; Cui, Y.; Hu, J.; Jia, W. Relation classification using coarse and fine-grained networks with SDP supervised key words selection. In Proceedings of the Knowledge Science, Engineering and Management: 11th International Conference, KSEM 2018, Changchun, China, August 17–19, 2018, Proceedings, Part I 11. Springer, 2018, pp. 514–522.
8. Xu, J.; Wang, Y. Enhancing Healthcare Recommendation Systems with a Multimodal LLMs-based MOE Architecture. *arXiv preprint arXiv:2412.11557* **2024**.
9. Jin, T. Integrated Machine Learning for Enhanced Supply Chain Risk Prediction **2025**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.