Mitigating Security Vulnerabilities in Open-Source Large Language Models via a Supervisor Agent Architecture

Öner Aytaş
Işık University, Turkey
oner.aytas@isikun.edu.tr

Mehmet Ali Bayram
Yıldız Technical University, Turkey
malibayram20@gmail.com

Tuğçe Şen
Işık University, Turkey
tugce.sen@isikun.edu.tr

Banu Diri
Yıldız Technical University, Turkey
diri@yildiz.edu.tr

Göksel Biricik
Yıldız Technical University, Turkey
gbiricik@yildiz.edu.tr

Abstract

This study proposes an independent Supervisor Agent approach based on a Turkish safety classifier to protect open-source large language models against harmful prompts. The proposed TurkGuard model classifies user prompts as Safe or Unsafe before they are forwarded to the main model and provides an additional security layer by blocking harmful prompts. The study addresses large language model security threats such as Prompt Injection and Jailbreaking and discusses the limitations of existing English-oriented safety solutions due to the morphological characteristics of Turkish. Experimental results show that the fine-tuned small-scale Qwen3.5-based TurkGuard model can provide high attack detection performance and a low false alarm rate for Turkish prompts.

Keywords: Large language models, open-source models, Turkish safety classification, Prompt Injection, Supervisor Agent, TurkGuard.

1 Introduction

Large Language Models (LLMs) have shown significant progress in natural language processing in recent years and have begun to be used in many different areas such as text generation, question-answering systems, code generation, and decision support systems. In particular, together with foundation models based on the Transformer architecture, it has become possible for a single model to perform a large number of tasks with high accuracy (Bommasani et al., 2021). Models such as GPT, Llama, Qwen, Gemma, Mistral, and DeepSeek are widely used not only in academic research but also in healthcare, education, finance, public-sector, and industrial applications (OpenAI, 2023).

In parallel with this development, open-source large language models have become an important alternative for researchers and developers. Open-source models provide advantages such as access to model weights, the ability to run within institutional systems, customizability, and low-cost deployment. Especially in applications where data privacy is critical, the ability to process user data without sharing it with third-party services has led to the preference for open-source models. However, this flexible structure also brings some important security risks. The modifiability of model weights, the possibility of re-fine-tuning, or the removal of safety restrictions make open-source models have a broader attack surface than closed-source models.

Today, large language models are used not only as systems that generate natural language but also as core components of intelligent systems that can interact with different software components. Thanks to capabilities such as making API calls, accessing databases, executing code, and using external tools, LLM-based applications can perform operations on real systems. While this situation expands the areas of use of the models, it also significantly increases the impact of security vulnerabilities. A malicious prompt may not only produce inappropriate text but may also lead to serious consequences such as the execution of system commands, the transfer of sensitive information to an external environment, or the performance of unauthorized actions.

With the widespread use of LLM-based systems, security threats have also diversified. In the OWASP Top 10 for LLM Applications document published by OWASP, threats such as Prompt Injection, Sensitive Information Disclosure, Excessive

Agency, Insecure Output Handling, and Training Data Poisoning are shown among the main security risks for large language models (OWASP Foundation, 2025). Unlike traditional software security vulnerabilities, these attacks target the model's semantic interpretation capability and often cannot be detected by classical security mechanisms.

In the literature, many studies have been conducted especially on Prompt Injection and Jailbreaking attacks. In Prompt Injection attacks, the attacker aims to make the model ignore system instructions through malicious instructions placed in user inputs or external data sources (Greshake et al., 2023). In Jailbreaking attacks, the aim is to bypass the model's safety policies by using role-playing, scenario construction, or multi-step reasoning techniques (Wei et al., 2023; Zou et al., 2023). In particular, fine-tuning performed on open-source models or the removal of safety filters may increase the probability of success of these attacks.

Various safety solutions such as Llama Guard, NeMo Guardrails, and Azure AI Content Safety have been developed for these security problems (Inan et al., 2024; NVIDIA, 2024; Microsoft, 2025). However, a significant portion of these solutions has been developed on English-oriented datasets and does not guarantee the same success in different languages, especially in agglutinative and morphologically rich languages such as Turkish. Turkish exhibits a more complex structure than English for safety classification because of its word derivation structure, the effect of suffixes on meaning, and context-sensitive language properties. Nevertheless, the number of studies conducted on the security of Turkish large language models is quite limited.

In this study, we propose a lightweight Supervisor Agent architecture designed to improve the security of open-source large language models against Prompt Injection attacks. The proposed architecture operates independently of the protected LLM by analyzing incoming user prompts before they reach the target model. To support this architecture, we develop TurkGuard, a lightweight Turkish safety classifier capable of performing binary Safe/Unsafe prompt classification. TurkGuard is designed to operate as an independent decision-making component and can be integrated with different open-source large language models without modifying their internal architecture.

Rather than introducing another standalone language model, this study presents a practical security framework in which TurkGuard functions as a lightweight Turkish safety classifier that enhances the security of existing open-source LLMs through an independent Supervisor Agent architecture.

1.1 Gap in the Literature and Motivation of the Study

In recent years, important studies have been carried out to improve the safety of large language models. In particular, solutions such as Llama Guard, NeMo Guardrails, and Azure AI Content Safety focus on the detection of harmful prompts, content filtering, and input-output validation mechanisms (Inan et al., 2024; NVIDIA, 2024; Microsoft, 2025). However, the vast majority of these solutions have been developed on English datasets and do not provide sufficient experimental evaluation for agglutinative and morphologically complex languages such as Turkish. In Turkish, the ability of suffixes to change meaning, context-dependent expressions, and different word derivation structures make safety classification more difficult than in English.

The broader adversarial NLP literature also shows that language-based systems can be manipulated not only through direct harmful prompts but also through subtle input transformations, universal triggers, hidden backdoors, and poisoned training data. Character-level adversarial attacks such as HotFlip demonstrate that small textual perturbations can substantially reduce classifier accuracy (Ebrahimi et al., 2018), while universal adversarial triggers show that short trigger sequences can cause systematic model failures across inputs (Wallace et al., 2019). Similarly, backdoor and poisoning studies indicate that language models and text classifiers can be manipulated through hidden triggers or poisoned examples without necessarily degrading clean-task performance (Dai et al., 2019; Li et al., 2021; Xu et al., 2021). These findings support the need for an external security layer that can evaluate prompts before they reach the main model.

Recent LLM safety research has further emphasized the importance of red-teaming, benchmark-based evaluation, and multilingual alignment. Red-teaming studies based on multi-turn or chained utterances show that harmful model behavior may emerge through interaction patterns rather than a single explicit malicious prompt (Bhardwaj & Poria, 2023). Safety datasets and benchmarks such as

BeaverTails and SALAD-Bench provide structured taxonomies for harmful instructions and model safety evaluation (Ji et al., 2023; Li et al., 2024). In addition, multilingual alignment studies report that safety behavior may vary across languages and that local linguistic preferences should be considered when reducing harmful outputs (Ahmadian et al., 2024; Friedrich et al., 2024). These studies are directly aligned with the motivation of the present work, which focuses on Turkish prompt safety classification.

Although there are many studies in the literature examining Prompt Injection and Jailbreaking attacks, research that systematically evaluates the safety performance of Turkish open-source large language models is quite limited. In particular, the lack of comprehensive studies in which different open-source models are compared under the same attack set makes it difficult to determine which models are more robust in terms of security.

In this context, in a study conducted by the authors of this paper, the safety performance of 35 different open-source large language models against Prompt Injection attacks was evaluated using 790 harmful prompts prepared in Turkish (Aytaş et al., 2026). The findings show that there are important differences among open-source models in terms of security. While some models achieved safety scores above 90%, in some models this value was observed to fall below 40%. This situation reveals that the safety levels of open-source large language models can vary significantly depending on model architecture, training data, and alignment processes.

Rank	Model	Safety Score
1	Qwen3.6:27B	98.23%
2	Cosmos Turkish-Gemma-9B	96.71%
3	Qwen3.5:27B	95.70%
4	Qwen3:30B	92.66%
5	Kizagan-E4B Turkish Reasoning	91.77%
...	...	...
31	KocDigital LLM-8B	49.37%
32	Kumru	46.46%
33	Cure-Med-14B	45.95%
34	Turkcell LLM-7B	39.49%
35	Commencis LLM	35.95%

Table 1: Open-source large language models with the highest and lowest safety scores in Turkish Prompt Injection tests (Aytaş et al., 2026)

These results show that there are serious performance differences among open-source models in terms of security. Although models with high safety levels exist, especially models with low safety scores may create important risks when used directly in real systems. In addition, even models with high safety scores are not completely immune to attacks such as Prompt Injection and Jailbreaking. Therefore, leaving security only to the model's own alignment mechanisms is not a sufficient solution.

The main motivation of this study is to develop an independent supervision mechanism that can improve the safety level of open-source large language models without changing the model architecture. For this purpose, a Supervisor Agent that operates in front of any open-source large language model was designed. The proposed structure classifies prompts received from users as Safe and Unsafe before they are forwarded to the main model and allows only prompts determined to be safe to reach the model. Thus, security supervision can be carried out independently of the model, and an additional protection layer that can be easily integrated with different open-source large language models is provided.

The main contributions of this study are summarized below:

An independent, general-purpose Supervisor Agent architecture is proposed as a decision-making security layer. This architecture operates completely decoupled from the target open-source large language model, providing an additional layer of protection against Prompt Injection and similar attacks by blocking harmful prompts before they reach the main model.

To power this architecture, TurkGuard, a lightweight and highly efficient safety classifier fine-tuned specifically to evaluate and classify user prompts as Safe or Unsafe, has been developed.

Comprehensive vulnerability assessments and benchmark tests were conducted. The evaluation focuses entirely on the Turkish language, utilizing an English comparative baseline to systematically measure the safety performance of various LLMs and demonstrate the real-world efficacy of the proposed defense mechanism.

2 Methodology

In this study, a two-stage security architecture based on a Supervisor Agent is proposed to protect open-source large language models against Prompt Injection and similar harmful prompts. The overall workflow of the proposed architecture is illus-

trated in Figure 1. In the proposed architecture, all prompts received from the user are first analyzed by the Supervisor Agent fine-tuned for security purposes. The Supervisor Agent semantically evaluates the prompt and classifies it as Safe or Unsafe. Prompts classified as safe are routed to the open-source large language model to be protected, while prompts evaluated as harmful are rejected before being forwarded to the main model. Thus, security supervision is performed by an independent decision mechanism rather than directly by the main model.

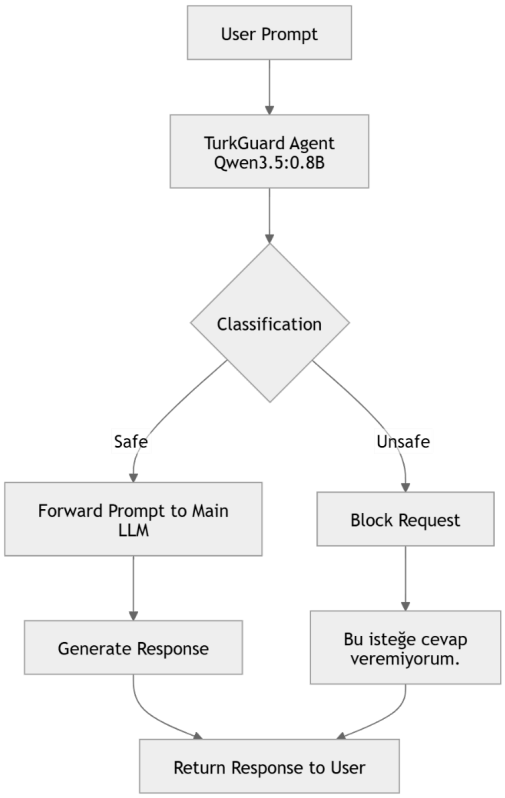

Figure 1: Proposed Supervisor Agent architecture.

The proposed architecture is designed to operate completely independently of the large language model to be protected. Therefore, it can be applied to any open-source large language model without requiring additional architectural changes. Separating the safety classifier from the main model enables the security layer to be reused across different models and integrated into existing systems with minimal changes.

2.1 Threat Model

In this study, it is assumed that the attacker can access the system only through user prompts. The attacker's goal is to bypass safety policies by using Prompt Injection, Jailbreaking, or similar social engineering techniques and to cause the main model to produce harmful content. In contrast, the attacker does not have direct access to the Supervisor Agent model, model weights, or system infrastructure.

The proposed approach aims to detect harmful prompts at an early stage through security analysis performed before the prompt reaches the model. Thus, the main model never encounters potentially harmful prompts, and the attack is intended to be blocked at the initial stage.

2.2 Supervisor Agent Fine-Tuning

Three low-parameter models, namely Qwen3.5-0.8B, LlamaGuard, and Gemma4 E2B, were considered as candidate Supervisor Agents. These models were selected because the aim of the proposed system is to create a lightweight front-end protection layer that consumes fewer computational resources than the protected LLM while still providing fast prompt-level security filtering. Among these alternatives, Qwen3.5-0.8B was selected as the base model for TurkGuard not only due to its low computational cost and fast inference time, but also because of its robust multilingual tokenizer and strong baseline comprehension of Turkish semantics compared with other sub-billion-parameter models. Therefore, Qwen3.5-0.8B provides a practical balance between efficiency, multilingual capability, and Turkish prompt understanding for the Supervisor Agent setting. The model was fine-tuned using a Sequence Classification approach not for text generation, but only for evaluating prompts in terms of security.

2.2.1 Dataset Construction

To train the proposed Turkish safety classifier, a balanced binary classification dataset was constructed by combining multiple Turkish-language resources. The final dataset consists of 51,264 labeled samples, including 25,843 Safe (50.41%) and 25,421 Unsafe (49.59%) instances.

The Unsafe class was created by combining three different sources. The first source consists of Turkish Prompt Injection prompts obtained from the Turkish Prompt Injection & Jailbreak Benchmark Dataset, which was compiled by translating and localizing publicly available English Prompt Injection and Jailbreak datasets into Turkish while incorporating Turkish cultural and linguistic characteristics (Aytaş, 2026). This dataset contains

translated, localized, and manually extended adversarial prompts collected from multiple benchmark sources. The second source consists of Prompt Injection responses generated during previous security analyses. The third source consists of a subset of manually labeled Turkish hate speech posts obtained from the publicly available turkish-hate-speech dataset on Hugging Face, which is based on the dataset introduced by Mayda et al. (2021). Selected samples from this resource were incorporated to increase the linguistic diversity of harmful expressions beyond Prompt Injection attacks.

The Safe class consists of benign Turkish prompts that do not contain malicious intent or harmful instructions. To prevent class imbalance, both classes were kept approximately equal in size. Duplicate records were removed, and all samples were mapped into a unified binary labeling scheme, namely Safe and Unsafe.

The resulting dataset was randomly divided into 85% training and 15% testing sets. Given the dataset size of 51,264 examples, the 15% split yields approximately 7,690 test instances, which provides a statistically robust evaluation set. Due to the high computational costs associated with fine-tuning LLMs, standard cross-validation was deemed computationally prohibitive; therefore, the 85/15 split was adopted as a practical strategy consistent with widely accepted deep learning practices for datasets of this scale. In addition, a subset of the training data was monitored internally as a validation set to prevent overfitting during the hyperparameter tuning phase. The previously developed pool of 790 Turkish Prompt Injection prompts was intentionally excluded from the training dataset and reserved exclusively for independent evaluation. This separation prevents data leakage and enables an unbiased assessment of the classifier's generalization capability.

As a result of the fine-tuning process, the developed Supervisor Agent performs binary classification for each prompt received from the user and produces only a Safe or Unsafe label. While prompts classified as Unsafe are blocked before being forwarded to the main model, prompts evaluated as Safe are routed to the open-source large language model to be protected.

2.3 Experimental Setup

To evaluate the success of the proposed architecture, an independent evaluation set consisting of 790 Turkish harmful prompts previously created by the researchers and containing different Prompt Injection techniques was used. All of these prompts were labeled as harmful, and the main task of the Supervisor Agent is to correctly classify these prompts as Unsafe. Since the test set was not used during the training process, it makes it possible to measure the model's real generalization performance.

To evaluate the false positive behavior of the Supervisor Agent, a Turkish multi-domain knowledge exam consisting 6200 questions was also used. This dataset consists of safe prompts representing daily usage scenarios. Thus, not only the model's ability to block harmful prompts but also its ability to correctly route normal user queries to the main model was evaluated.

Thanks to these two independent evaluation sets, the proposed system was analyzed both in terms of attack detection success and decision-making performance in normal usage scenarios.

3 Experimental Results

The performance of the proposed Supervisor Agent was evaluated under three different experimental scenarios.

First, the base Qwen3.5-0.8B model used without fine-tuning was compared with the proposed fine-tuned Supervisor Agent. Thus, the contribution of the fine-tuning process to safety performance was demonstrated.

Second, the proposed method was compared with Llama Guard, which is widely used in the literature, and the Gemma4-2B fine-tuned model developed on a different open-source model. The purpose of this comparison is to compare the success of the proposed approach on Turkish Prompt Injection attacks with existing safety solutions.

Finally, the TR-MMLU dataset, consisting 6200 questions, was used to evaluate whether the Supervisor Agent developed for security purposes incorrectly blocks normal user prompts. TR-MMLU is a comprehensive assessment dataset developed to evaluate the Turkish language performance of large language models and contains Turkish multiple-choice questions from different disciplines (Bayram et al., 2024). Thus, the proposed system was evaluated both in terms of its success in detecting harmful prompts and its behavior in normal usage scenarios.

3.1 Prompt Injection Detection Performance

Using the developed Turkish safety dataset, fine-tuning experiments were conducted on three candidate low-parameter models: Llama Guard, Gemma4-2B, and Qwen3.5-0.8B. The comparative results of these fine-tuned models, together with the non-fine-tuned Qwen3.5-0.8B baseline, are presented in Table 2.

The attack detection performance of the Supervisor Agent was evaluated on 790 Turkish Prompt Injection prompts previously created by the researchers and not used during training. All prompts in the test set were labeled as harmful (Unsafe). Therefore, the evaluation was carried out mainly based on the Recall (Detection Rate) metric.

Model	Fine-tune	TP	FN	Recall (%)
Llama Guard	Applied	178	612	22.53
Gemma4-2B	Applied	442	348	55.95
Qwen3.5-0.8B (Base)	Not Applied	400	390	50.63
Qwen3.5-0.8B (TurkGuard)	Applied	577	213	73.04

Table 2: Comparison of different models on Turkish Prompt Injection prompts.

When Table 2 is examined, it is seen that the base Qwen3.5-0.8B model without fine-tuning can correctly detect only 50.63% of harmful prompts. This result shows that directly using general-purpose large language models as safety classifiers is not sufficient.

The fine-tuning study carried out on the Gemma4-2B model provided a limited performance increase compared with the base model (55%), but the proposed Supervisor Agent architecture achieved the highest performance with 73.04% detection success. In particular, the very low success of the Llama Guard model on Turkish prompts (24%) shows that English-centered safety models are not sufficient for direct use in morphologically rich languages such as Turkish.

These results reveal that training the safety classifier with target-language-specific datasets provides an important advantage in detecting Prompt Injection attacks.

3.2 Performance in Normal Usage Scenarios

It is not sufficient for a security system only to block harmful prompts. At the same time, it is expected not to incorrectly classify normal user queries as harmful. For this reason, the proposed Supervisor Agent was also evaluated on the TR-MMLU dataset, which was developed in the authors' previous work and consists 6200 Turkish multiple-choice questions (Bayram et al., 2024).

Dataset	Total	Safe	Unsafe	Safe Acc. (%)
TR-MMLU	6200	5938	262	95.77

Table 3: Evaluation on the TR-MMLU dataset.

The results show that the proposed Supervisor Agent correctly classifies approximately 95% of normal informational prompts as Safe. This finding shows that the model not only detects attacks but can also operate with a low false alarm rate in daily usage scenarios.

4 Discussion

The experimental results show that a fine-tuned small-scale safety model can perform significantly more successful safety classification than a general-purpose base model. It is particularly noteworthy that the Qwen3.5-based Supervisor Agent (TurkGuard), which has only 0.8 billion parameters, performs better than some safety models with larger numbers of parameters. This situation shows that task-oriented fine-tuning may be more decisive than model size in safety tasks.

In addition, the low performance of Llama Guard on Turkish prompts can be considered a natural result of the fact that existing safety solutions have largely been developed on English datasets. The agglutinative structure of Turkish, its context-sensitive expressions, and its rich morphological properties can limit the generalization success of English-centered safety models. This finding reveals the importance of developing language-specific safety models for different languages.

The evaluation carried out on the TR-MMLU dataset shows that the proposed approach does not focus only on detecting harmful prompts but can also classify normal user queries as Safe with high accuracy. The approximately 95% correct classification success is important because it shows that the proposed structure can operate with a low false alarm rate in real usage scenarios.

Another important output of this study is that the proposed Supervisor Agent offers a general security layer that can be used not only in newly developed systems but also to increase the security levels of existing open-source large language models. In the security evaluations conducted in our previous study, it was shown that some open-source models had quite low safety scores against Prompt

Injection attacks (Aytaş et al., 2026). When the architecture proposed in this study is placed in front of these models as an independent security layer, it can filter a significant portion of harmful prompts without changing the architecture or weights of the main model. Thus, it may be possible to make open-source large language models with low safety scores more secure, and a practical security improvement can also be achieved without the need to retrain existing models.

However, the proposed approach does not aim to provide absolute protection against all types of attacks. In particular, additional studies are needed for multi-step attacks, indirect Prompt Injection techniques, and new attack methods that may be developed in the future. In future studies, more comprehensive experiments on different open-source models, the development of multilingual safety classifiers, and the integration of dynamic safety policies into the Supervisor Agent architecture are planned.

5 Limitations

A notable limitation of this study relates to the evaluation of false positive rates using the TR-MMLU dataset. Since TR-MMLU primarily consists of multiple-choice academic questions, it does not fully capture the complex, conversational, and diverse prompt distributions encountered in real-world deployments. Consequently, the 95.77% safe accuracy achieved in normal usage scenarios may overestimate the model's actual deployment performance. Future studies should evaluate false positive behavior on real-world, conversational Turkish prompt datasets.

References

Ahmadian, A., Ermis, B., Goldfarb-Tarrant, S., Kreutzer, J., Fadaee, M., & Hooker, S. (2024). The multilingual alignment prism: Aligning global and local preferences to reduce harm. arXiv. https://arxiv.org/abs/2406.18682

Aytaş, Ö. (2026). TurkGuard: A fine-tuned Turkish safety classifier based on Qwen3.5-0.8B (Version 1.0) [Large language model]. Hugging Face. Hugging Face Model Repository. https://huggingface.co/OnerAYTAS/TurkGuard_Qwen3.5B_0.8B

Aytaş, Ö.; Şen, T.; Diri, B.; Biricik, G.; Bayram, M.A. (2026)Benchmarking Prompt Injection Attacks on LLMs: Turkish Vulnerability Assessment and English Comparative Analysis. Applied Sciences 2026, 16(13), 6740. https://doi.org/10.3390/app16136740

Aytaş, Ö. (2026). Turkish Prompt Injection & Jailbreak Benchmark Dataset [Data set]. Hugging Face. Hugging Face Dataset Repository.

Bayram, M. A., Fincan, A. A., Gümüş, A. S., Diri, B., Yıldırım, S., & Aytaş, Ö. (2024). Setting standards in Turkish NLP: TR-MMLU for large language model evaluation. arXiv preprint arXiv:2501.00593.

Bhardwaj, R., & Poria, S. (2023). Red-teaming large language models using chain of utterances for safety-alignment. arXiv. https://arxiv.org/abs/2308.09662

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. Stanford Center for Research on Foundation Models. https://arxiv.org/abs/2108.07258

Dai, J., Chen, C., & Li, Y. (2019). A backdoor attack against LSTM-based text classification systems. IEEE Access, 7, 138872–138878. https://doi.org/10.1109/ACCESS.2019.2941376

Ebrahimi, J., Rao, A., Lowd, D., & Dou, D. (2018). HotFlip: White-box adversarial examples for text classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL).

Friedrich, F., Tedeschi, S., Schramowski, P., Brack, M., Navigli, R., Nguyen, H., Li, B., & Kersting, K. (2024). LLMs lost in translation: M-ALERT uncovers cross-linguistic safety inconsistencies. arXiv. https://arxiv.org/abs/2412.15035

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Fritz, M., & Tramèr, F. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection attacks. arXiv. https://arxiv.org/abs/2302.12173

Inan, H., Upasani, K., Wang, C., et al. (2024). Llama Guard: LLM-based input-output safeguard for human-AI conversations. arXiv. https://arxiv.org/abs/2312.06674

Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., Zhang, C., Sun, R., Wang, Y., & Yang, Y. (2023). BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. arXiv. https://arxiv.org/abs/2307.04657

Li, L., Dong, B., Wang, R., Hu, X., Zuo, W., Lin, D., Qiao, Y., & Shao, J. (2024). SALAD-Bench: A hierarchical and comprehensive safety benchmark for large language models. arXiv. https://arxiv.org/abs/2402.05044

Li, S., Liu, H., Dong, T., Zhao, B. Z. H., Xue, M., Zhu, H., & Lu, J. (2021). Hidden backdoors in human-centric language models. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 1–18. https://doi.org/10.1145/1122445.1122456

Mayda, İ., Demir, Y. E., Dalyan, T., & Diri, B. (2021). Hate Speech Dataset from Turkish Tweets. In 2021 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1–4). IEEE. https://doi.org/10.1109/ASYU52992.2021.9599042

Microsoft. (2025). Azure AI Content Safety Documentation. https://learn.microsoft.com/azure/ai-services/content-safety/

NVIDIA. (2024). NeMo Guardrails Documentation. https://docs.nvidia.com/nemo/guardrails/

OpenAI. (2023). GPT-4 technical report. arXiv. https://arxiv.org/abs/2303.08774

OWASP Foundation. (2025). OWASP Top 10 for Large Language Model Applications. https://genai.owasp.org/

Wallace, E., Feng, S., Kandpal, N., Gardner, M., & Singh, S. (2019). Universal adversarial triggers for attacking and analyzing NLP. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Wei, A., Haghtalab, N., & Steinhardt, J. (2023). Jailbroken: How does LLM safety training fail? Advances in Neural Information Processing Systems, 36.

Xu, C., Wang, J., Tang, Y., Guzmán, F., Rubinstein, B. I., & Cohn, T. (2021). A targeted attack on black-box neural machine translation with parallel data poisoning. In Proceedings of the Web Conference 2021, 1–12. https://doi.org/10.1145/3442381.3450034

Zou, A., Wang, Z., Kolter, J. Z., & Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. arXiv. https://arxiv.org/abs/2307.15043