UDC 004.8:004.056

# Method of Counteracting Manipulative Queries to Large Language Models

Yehor Kovalchuk[1], Mykhailo Kolomytsev[1]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Institute of Physics and Technology*

**Abstract**

The integration of Large Language Models (LLMs) into critical infrastructure (SIEM, SOAR) has introduced new attack vectors, specifically prompt injection and jailbreaking. Traditional defense mechanisms, such as input sanitization and Reinforcement Learning from Human Feedback (RLHF), often fail against semantic obfuscation and indirect injections due to their inability to distinguish between control instructions and data context. This paper proposes a novel method for detecting manipulative prompts based on a Multi-Head DistilBERT architecture. Unlike standard binary classifiers, the proposed model decomposes the detection task into four semantic vectors: malicious intent, instruction override, persona adoption, and high-risk action. To address the scarcity of labeled adversarial datasets, we implemented a hybrid data generation strategy using Knowledge Distillation, employing a superior model (Teacher) to label synthetic attacks for the compact Student model. Experimental results on both synthetic and real-world datasets demonstrate that the proposed system achieves a Recall of 0.99, significantly outperforming traditional TF-IDF and keyword-based baselines. The solution operates effectively as a middleware layer, ensuring real-time protection with low computational latency suitable for deployment on edge devices.

*Keywords*: Large Language Models, Prompt Injection, Jailbreaking, NLP Security, DistilBERT, Adversarial Machine Learning.

## Introduction

The rapid integration of Large Language Models (LLMs), such as GPT-5, Claude, and Llama, into critical information systems has fundamentally transformed the cyber threat landscape. These models are no longer passive text generators but serve as the backbone for corporate assistants, Security Information and Event Management (SIEM) copilots, and autonomous agents capable of executing API calls. However, this utility comes with a significant architectural vulnerability inherent to the Transformer architecture: the mechanism of Self-Attention does not natively distinguish between system instructions (control plane) and user input (data plane) [1].

This lack of context isolation has given rise to a new class of attacks known as Prompt Injection and Jailbreaking, where adversaries manipulate the model's output by injecting malicious instructions that override safety guardrails. While direct injections via user interfaces are well-documented, the emergence of Retrieval-Augmented Generation (RAG) systems has exacerbated the risk through Indirect Prompt Injection. In this scenario, an LLM processing external data (e.g., email logs or websites) ingests a hidden payload that forces the model to execute unauthorized actions, effectively turning the LLM into a confused deputy [2, 4, 16].

Current defense mechanisms remain insufficient against these semantic threats. Traditional input sanitization and keyword filtering are fundamentally brittle; they operate on a lexical level and are easily bypassed by obfuscation techniques, such as token fragmentation or base64 encoding, which rely on the "tokenization mismatch" between the filter and the LLM [5, 6]. Furthermore, safety measures embedded during training, such as Reinforcement Learning from Human Feedback (RLHF), are reactive by nature. They defend only against attack patterns present in the training distribution, leaving models vulnerable to zero-day semantic manipulations and complex social engineering vectors.

To address these limitations, this paper presents a proactive method for detecting manipulative prompts using a **Multi-Head**

**DistilBERT** architecture. Unlike standard binary classifiers, our approach decomposes the detection task into specific structural violations: malicious intent, instruction override, persona adoption, and high-risk actions. By analyzing the semantic structure rather than mere keywords, the system acts as a middleware layer capable of identifying obfuscated attacks in real-time. Furthermore, we introduce a hybrid data generation strategy using Knowledge Distillation, leveraging a superior model (GPT-5) to automatically label complex attack vectors for the compact student model, ensuring robustness against evolving threats.

## 1. Proposed Method

The analysis of existing models [9, 11, 12, 13] showed their low effectiveness in detecting manipulative prompts.

To address the limitations of reactive defense mechanisms, we propose a proactive middleware architecture designed to detect manipulative prompts in real-time. Unlike traditional approaches that rely on keyword filtering or generic binary classification, our method utilizes deep semantic analysis to decompose the structure of a prompt. This approach allows for the differentiation between legitimate data context and malicious control instructions, even when obfuscation techniques are employed.

### 1.1. Multi-head Classification Architecture

The core of the proposed system is built upon the DistilBERT model [3]. This architecture was selected to balance the need for deep semantic understanding (via the Transformer self-attention mechanism) with the low-latency requirements of real-time cybersecurity systems, making it suitable for deployment on resource-constrained hardware.

**DistilBERT-based Multi-Task Learning Design**

We modified the standard DistilBERT architecture by replacing the single output layer with a Multi-Task Learning (MTL) configuration (figure 1). The model features four independent fully connected layers ("heads"), each responsible for detecting a specific structural aspect of an attack:

- **Head 1: Malicious Intent.** Determines the overall probability that the prompt contains malicious content.
- **Head 2: Override Instruction.** Specifically detects attempts to negate or rewrite the system prompt (e.g., "Ignore previous instructions").
- **Head 3: Persona Adoption.** Identifies attempts to force the model into a specific role that bypasses safety guidelines (e.g., "Act as DAN").
- **Head 4: High-Risk Action.** Detects semantic patterns related to dangerous execution capabilities, even if the language is veiled.

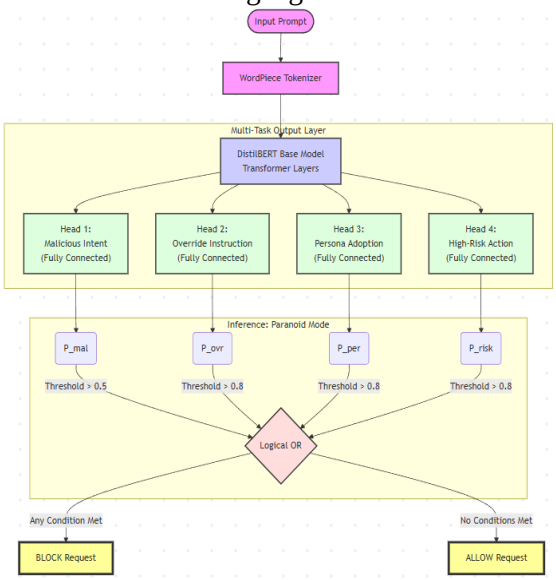


**Figure 1**. Architecture of the Multi-Head DistilBERT Classifier

The model is trained by minimizing a combined loss function, defined as follows:

$$L_{total} = L_{malicious} + \lambda \sum_{i \in \{ovr,per,risk\}}^{n} \mathrm{L_i} \quad (1)$$

where $\lambda = 0.5$ is an empirically selected coefficient used to balance the contribution of the auxiliary heads to the total loss.

**Ensemble Decision Logic and "Paranoid Mode"**

In cybersecurity contexts, a False Negative (missed attack) typically carries a much higher cost than a False Positive (blocked legitimate request). To address this, we implemented an ensemble decision logic referred to as "Paranoid Mode" during the inference stage.

Rather than averaging the outputs, the system triggers a block if any of the specialized heads detect an anomaly with high confidence. The thresholds for the auxiliary heads were set to 0.8 based on ROC curve analysis of the validation set to minimize false positives on complex but legitimate prompts.

### 1.2. Hybrid Data Generation Strategy

A significant challenge in training robust detectors for Large Language Models (LLMs) is the scarcity of high-quality, granularly labeled datasets that distinguish between specific attack vectors (e.g., persona adoption vs. direct injection). To overcome this, we developed a hybrid strategy combining synthetic data generation with automated labeling (figure 2).

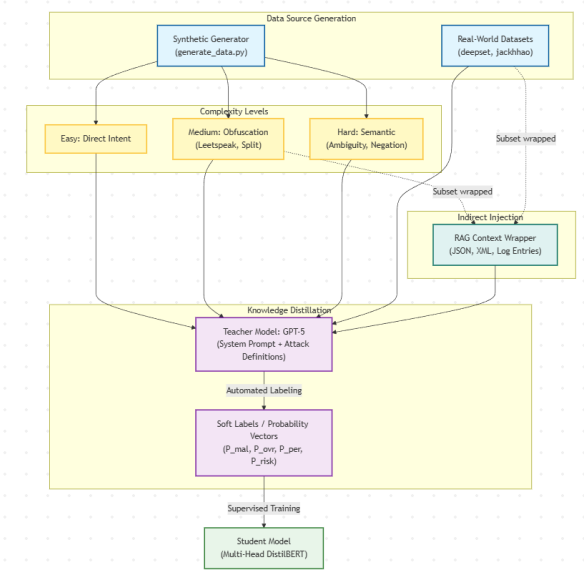


**Figure 2.** Hybrid Data Generation and Labeling Pipeline

**Synthetic Data Augmentation and Obfuscation**

We developed a specialized data generator to create attack samples across three difficulty levels, ensuring the model learns to recognize adversarial intent rather than just keywords:

- **Easy (Direct Intent):** Unobfuscated malicious requests (e.g., "Write a phishing email") used to establish baseline detection capabilities.
- **Medium (Obfuscation):** Prompts modified using algorithmic obfuscation techniques to bypass keyword filters. This includes splitting words (e.g., "h-a-c-k"), leetspeak substitutions (e.g., "h4ck"), and noise injection (e.g., "h#ack").
- **Hard (Semantic Manipulation):** Complex prompts utilizing linguistic ambiguity and negation to test context understanding (e.g., distinguishing "I do not want to kill the process" from malicious commands).

A total of 3,000 synthetic examples were generated, balanced across these classes.

**Automated Labeling via Knowledge Distillation**

To enable the compact DistilBERT model to recognize complex patterns found in real-world attacks, we applied a Knowledge Distillation approach. We utilized existing binary-labeled datasets (such as `deepset/prompt-injections` and `jackhhao/jailbreak-classification`) and processed them through a superior "Teacher" model (GPT-5).

Using a custom system prompt containing the definitions of specific attack patterns (Instruction Conflict, Dangerous Persona, etc.), the Teacher model analyzed each sample and generated detailed probability scores for the auxiliary heads. Additionally, to simulate Indirect Prompt Injection threats in RAG systems, a subset of malicious prompts was automatically wrapped in data structures (JSON, XML, logs) to mimic context contamination scenarios.

## 2. Experimental Results and Discussion

To validate the effectiveness of the proposed Multi-Head DistilBERT model, we conducted a series of comparative experiments against traditional text classification baselines. The primary objective was to evaluate the model's resilience to obfuscation and its ability to generalize to real-world attack vectors.

### 2.1. Datasets and Baselines

The experiments utilized the hybrid dataset described in Section 1.2, comprising 3,000 synthetic samples (balanced across Easy, Medium, and Hard complexity levels) and a 20% holdout set from the deepset/prompt-injections dataset to represent real-world distribution.

We compared our proposed architecture against two industry-standard baselines:

- **Keyword Matching (RegEx):** A deterministic filter based on a blacklist of 200+ common malicious keywords (e.g., "ignore", "hack", "payload").
- **TF-IDF + Logistic Regression:** A classic statistical Machine Learning approach often used for spam detection, representing a non-contextual baseline.

### 2.2. Evaluation Metrics

While we tracked Accuracy and F1-Score, the primary metric for evaluation was **Recall**. In the context of critical infrastructure protection (e.g., preventing Prompt Injection in a SIEM), a False Negative (missing an attack) poses a catastrophic risk, whereas a False Positive (blocking a benign query) is a manageable inconvenience. Therefore, our optimization goal was to maximize Recall.

### 2.3. Performance Analysis

#### Robustness Against Semantic Obfuscation

The comparative analysis on synthetic data revealed significant disparities in handling obfuscated and semantic attacks (figure 3).

On the **Synthetic Medium** dataset (obfuscation via token splitting and leetspeak), the Keyword Baseline performance collapsed, achieving an F1-Score of only **0.27**. This confirms that lexical filters are rendered ineffective by simple tokenization manipulations (e.g., "b-o-m-b"). In contrast, the proposed DistilBERT model, leveraging sub-word tokenization and contextual embeddings, maintained a high F1-Score of **0.98**.

On the **Synthetic Hard** dataset (semantic ambiguity and negation), the TF-IDF baseline struggled, achieving an F1-Score of **0.60**. The statistical approach failed to distinguish between safe contexts (e.g., "kill the process") and malicious intents (e.g., "kill the boss") due to its inability to capture word order and dependencies. The proposed Multi-Head architecture successfully resolved these ambiguities, achieving an F1-Score of **0.87**.

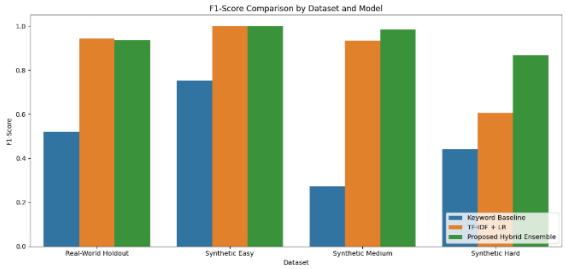


**Figure 3.** Comparative Analysis of F1-Scores across different datasets

#### Detection Capabilities on Real-World Attacks

The final evaluation on the real-world holdout dataset demonstrated the efficacy of the "Paranoid Mode" ensemble logic (figure 4). As shown in the comparison below, the proposed method achieved a **Recall of 0.99**, significantly outperforming the baselines.

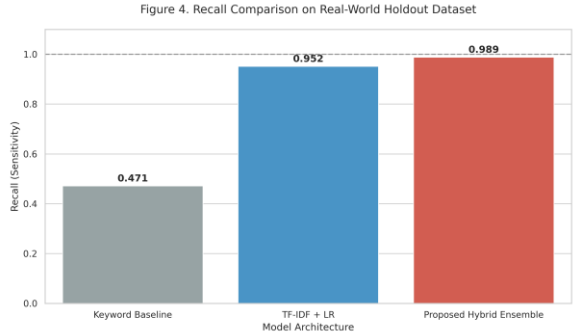


**Figure 4.** Recall Comparison on Real-World Holdout Dataset

While the TF-IDF model achieved a slightly higher Precision, it missed approximately 5% of attacks (Recall ~0.95). The proposed system missed less than 1% of attacks. The slight reduction in Precision (0.89 for the proposed method vs. 0.94 for TF-IDF) is a deliberate trade-off resulting from the aggressive multi-head aggregation strategy, ensuring that ambiguous prompts are blocked rather than allowed.

## Conclusions

This study addresses the critical security gap in the deployment of Large Language Models within corporate infrastructure, specifically targeting the vulnerability of Transformer architectures to semantic manipulation and indirect prompt injections. Our analysis confirmed that traditional defense mechanisms, such as input sanitization and reinforcement learning alignment (RLHF), are insufficient

against attacks that exploit the lack of context isolation between control instructions and data.

To mitigate these risks, we proposed and validated a novel detection system based on a Multi-Head DistilBERT architecture. By decomposing the classification task into four distinct semantic vectors—malicious intent, instruction override, persona adoption, and high-risk actions—our model successfully approximates the structural analysis of prompts. The integration of an ensemble "Paranoid Mode" logic ensured a high sensitivity to potential threats, achieving a Recall rate of **0.99** on real-world attack datasets.

Furthermore, the introduction of a hybrid data generation strategy, utilizing Knowledge Distillation from a superior teacher model (GPT-5), proved effective in overcoming the scarcity of labeled adversarial data. This approach enabled the compact student model to learn complex, non-linear attack patterns and resist obfuscation techniques that bypass standard lexical filters. The resulting middleware solution offers a robust, low-latency defense layer suitable for real-time protection of SIEM and SOAR systems against emerging adversarial NLP threats

## References


[1] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. 2019 Conf. North American Chapter Assoc. Comput. Linguistics, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[3] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[4] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications," Version 1.1, 2023. [Online]. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications/.

[5] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and Transferable Adversarial Attacks on Aligned Language Models," *arXiv preprint arXiv:2307.15043*, 2023.

[6] K. Greshake *et al.*, "Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," in Proc. 16th ACM Workshop on Artificial Intelligence and Security, 2023, pp. 79–90.

[7] Y. Bai *et al.*, "Constitutional AI: Harmlessness from AI Feedback," *arXiv preprint arXiv:2212.08073*, 2022.

[8] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How Does LLM Safety Training Fail?" in *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[9] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[10] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.

[12] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in Proc. 2020 Conf. Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[13] T. Brown *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[14] N. Carlini *et al.*, "Extracting Training Data from Large Language Models," in Proc. 30th USENIX Security Symposium, 2021, pp. 2633–2650.

[15] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] Y. Liu, G. Deng, Z. Xu, *et al.*, "Prompt Injection attack against LLM-integrated Applications," *arXiv preprint arXiv:2306.05499*, 2023.