Prompt Injection Detection via Key Feature Integration in Large Language Models

Seyeon Won
School of Computer Science and Engineering
Yeungnam University
Gyeongsan, Republic of Korea
seyeon@yu.ac.kr

Wooguil Pak
School of Computer Science and Engineering
Yeungnam University
Gyeongsan, Republic of Korea
wooguilpak@yu.ac.kr

Abstract— **This study proposes Total-Layer Detection (TLD), a novel framework that integrally utilizes all layers of a Large Language Model to detect prompt injection attacks effectively. To address the computational inefficiency arising from high-dimensional data, we selectively utilized only the core features decisive for attack identification. Experiments demonstrated that TLD achieves higher accuracy and superior efficiency compared to single-layer approaches by efficiently reducing the feature dimensionality from the concatenated total-layer vector while enhancing detection precision.**

***Keywords*— Large Language Models (LLMs), Prompt Injection, Total-Layer Detection, Feature Selection**

I. INTRODUCTION

Large Language Models (LLMs) demonstrate capabilities beyond simple text generation, performing complex tasks and achieving remarkable results across diverse fields [1]. While this expansion has maximized the utility of LLMs, it has simultaneously introduced new security threats, such as prompt injection [2, 3, 4]. Attackers inject malicious commands to disable the model's system prompts or bypass safety policies to induce harmful actions. Moreover, as attack techniques leveraging external data referenced by LLMs become known, interest in these security threats is intensifying.

To address these threats, detection methods that analyze a model's internal information are being researched. Notably, Layer Enhanced Classification (LEC) [5] research showed that attack detection is possible using information from a single layer alone. However, relying on a single layer has a limit. As the model becomes deeper, it fails to comprehensively capture the inference information distributed across layers.

To overcome this limitation, this study proposes a new detection framework, Total-Layer Detection (TLD), which integrally utilizes information from all layers of the LLM. To address the inefficiency that arises from simply connecting all information, we select only the key features critical for attack identification.

The main contributions of this paper are as follows:

- Integrated utilization of total-layer information: Captures the entire inference process of the model, not just a single layer, enabling precise detection of covertly hidden attack intentions.
- Efficient Feature Selection: We demonstrate efficiency by compressing the entire feature set through feature importance analysis while maintaining or improving detection accuracy.

II. RELATED WORK

This section reviews recent trends in prompt injection attacks threatening the safety of LLMs and examines existing research on defenses leveraging the model's internal information. First, we define prompt injection, then focus on analyzing the LEC, a representative example of internal information-based detection and the primary comparison target in this study.

A. Prompt Injection

Prompt injection is an attack method where an attacker injects malicious input to manipulate an LLM into ignoring predefined system prompts and performing actions contrary to the developer's intent. It is generally divided into direct prompt injection and indirect prompt injection. Direct prompt injection is a method that attempts to attack through the prompt input to the LLM. Known methods include utilizing explicit phrases like 'Ignore previous instructions' to induce the model to disregard the system prompt [2] or inferring the structure of the system prompt and combining it with an attack sentence [3]. Indirect prompt injection is a method that attempts attacks by manipulating external data referenced by LLMs without requiring direct malicious input from the user, leading to threats such as information leakage and remote manipulation [4].

B. Defense Techniques Utilizing Internal Model Information

One approach to detecting prompt injection leverages the model's internal information. LEC utilizes the LLM itself as a feature extractor, capitalizing on the fact that intermediate layers of LLM often exhibit high performance in embedding classification tasks. LEC evaluates classification performance for all layers, selects the single intermediate layer with the best performance, and then applies a pruning technique. This enables significantly lighter and more powerful detection performance. However, due to its structural reliance on information from a single layer, it has limitations in complex attack scenarios where information is distributed across

multiple layers or exhibits evasive patterns, potentially leading to reduced detection performance.

III. METHOD

This paper proposes a TLD framework that comprehensively utilizes information from all layers of an LLM, rather than focusing on specific layers, to detect prompt injections. Specifically, to address computational cost issues arising from high-dimensional data processing and maximize detection accuracy, we introduce an approach that selects and uses only key information through feature importance analysis.

A. Overview

The proposed system consists of three main stages. First, the Total-layer Hidden State Extraction stage captures all inference processes from shallow to deep layers of the model. Second, the feature importance analysis stage reduces dimensions by selecting only features critical for attack detection from the vast amount of information. Finally, the Prompt Injection Detection stage determines the final attack status based on the selected feature vectors. This approach enhances detection performance by utilizing information from all layers while removing unnecessary noise, enabling efficient and precise detection. The overall architecture of this TLD is illustrated in Fig. 1.

B. Total-Layer Hidden States Extraction

LLMs generate hidden states at each layer while processing input sequences. Since each layer captures distinct linguistic and contextual features, this study collects all hidden states generated by the model from the first layer to the last. The vectors from each collected layer are concatenated to form a feature vector containing the model's overall inference information for that input. While this vector holds rich information, its high dimensionality poses limitations: it incurs high computational costs for real-time processing and may contain unnecessary features or noise for analysis.

C. Feature Importance Analysis

Feature importance analysis is performed to select features that substantially contribute to detecting prompt injection attacks within the existing high-dimensional feature vectors. Features are sorted in descending order based on importance scores, and only the top K features are selected. This process extracts only information critical for attack identification, significantly reducing data dimensions and maximizing computational efficiency.

D. Prompt Injection Detection

The final selected feature vector is input into the detection classifier. This classifier analyzes the input feature vector to determine whether the input prompt is an attack or normal behavior. The classifier utilizes only the core information from all hidden layers, enabling high performance with low computational complexity.

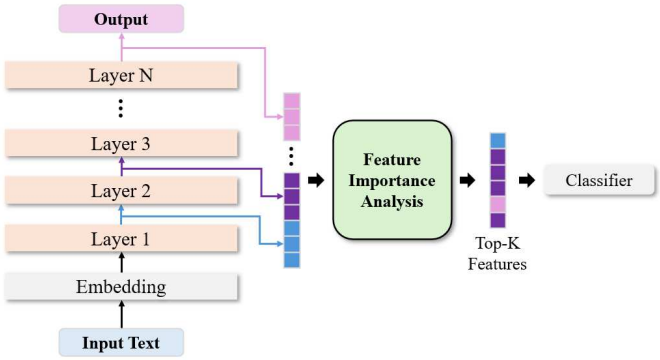

Figure 1. The Structure of TLD.

IV. EXPERIMENTS

This section validates the effectiveness of the proposed method, TLD. Experiments were conducted using various datasets and models, with a particular focus on analyzing the effects of utilizing total-layer information and feature selection compared to using only single-layer information.

A. Dataset

This study utilized two datasets: SPML [6], LMSYS+SALAD [7, 8]. Each dataset was equally split into training (3,300 samples), validation (1,700 samples), and test (1,700 samples) sets.

1) SPML: This dataset simulates real chatbot interaction environments, consisting of pairs of system prompts and user prompts. It defines attempts by users to intentionally violate or circumvent safety rules specified in system prompts as 'attacks', making it suitable for evaluating the ability to discern the inherent intent behind attacks.

2) LMSYS+SALAD: LMSYS is a dataset containing 1 million real conversation logs [7]. To ensure data quality, only logs deemed 'harmless' by the OpenAI Moderation API criteria, in English, and single-turn data were filtered and used as normal data. SALAD is a hierarchical safety benchmark dataset; only the base set subset containing harmful data was used as attack data [8]. Experiments were configured by combining both datasets at an equal ratio.

B. Feature Extraction Model & Classifier

Two pre-trained models, Qwen2.5-0.5B-Instruct and Qwen2.5-14B-Instruct [2], were used as feature extractors. Qwen2.5-0.5B-Instruct has 24 layers and an 896-dimensional hidden state, while Qwen2.5-14B-Instruct has 48 layers and a 5,120-dimensional hidden state. The final classifier for determining attack presence used the PyTorch torch.nn.Linear module. The input dimension was set to K, the number of selected features, and the output dimension was set to 2.

C. Comparison Targets

To demonstrate the validity and contribution of the proposed methodology, performance was compared against

TABLE I. LMSYS + SALAD DATASET EVALUATION RESULT

Feature Extraction Model	Method	Classification Layer (1~)	Number of Selected Features	Accuracy (%)	F1 (%)
Qwen2.5-0.5B	Feature Select	-	16,000	93.12%	93.12%
	Best Layer	7	896	**95.00%**	**95.00%**
	Last Layer	24	896	87.00%	86.99%
Qwen2.5-14B	Feature Select	-	700	**97.59%**	**97.59%**
	Best Layer	9	5,120	97.12%	97.12%
	Last Layer	48	5,120	94.47%	94.47%

TABLE II. SPML DATASET EVALUATION RESULT

Feature Extraction Model	Method	Classification Layer (1~)	Number of Selected Features	Accuracy (%)	F1 (%)
Qwen2.5-0.5B	Feature Select	-	17,000	**96.65%**	**96.65%**
	Best Layer	13	896	95.82%	95.82%
	Last Layer	24	896	89.82%	89.82%
Qwen2.5-14B	Feature Select	-	15,000	**99.18%**	**99.18%**
	Best Layer	27	5,120	99.06%	99.06%
	Last Layer	48	5,120	97.53%	97.53%

two main single-layer comparison groups. The Last Layer group uses only the hidden states extracted from the final hidden layer of the feature extraction model as features. The Best Layer group corresponds to the LEC methodology, which classifies using the single hidden layer within the feature extraction model that exhibits the best classification performance. Both groups used *torch.nn.Linear* as the classifier.

D. Experimental Results

Evaluation results for the LMSYS+SALAD, SPML datasets are presented in Table I and Table II, respectively. The proposed method outperformed single-layer approaches in three out of four scenarios. For LMSYS+SALAD (Table I), the proposed method slightly underperformed the Best Layer on the Qwen2.5-0.5B model but outperformed it on the larger Qwen2.5-14B model. Conversely, on the SPML (Table II) datasets, the proposed method recorded higher overall accuracy and F1-Score compared to Best Layer, regardless of model size.

Particularly in terms of feature selection efficiency, examining the number of selected features by the proposed method reveals that Qwen2.5-0.5B utilized an average of approximately 76.7% of the total features, whereas Qwen2.5-14B used an average of only 3.19% of the total dimensions. Notably, the Qwen2.5-14B result in Table I achieved higher accuracy using only 700 features—a number significantly fewer than the single-layer dimension (5,120)—suggesting that the proposed feature selection technique enables highly efficient and precise detection.

V. CONCLUSION

In this study, we propose a novel framework, TLD, to effectively detect prompt injection attacks threatening the safety of LLMs by integrally utilizing information from all layers of the model. Particularly, to solve the computational inefficiency problem arising from high-dimensional data when using total-layer information, we applied an approach that selectively utilizes only the core features decisive for attack identification.

The experimental results demonstrate that TLD achieves performance that is superior or at least comparable to existing approaches that rely solely on single-layer information. Notably, in experiments using the Qwen2.5-14B model, the proposed method achieved higher accuracy than the Best Layer approach using only 700 selected features—significantly fewer than the dimension of the hidden layer.

This study demonstrates the potential of attack detection techniques through the integration of total-layer information. However, the proposed method still has the limitation of requiring a relatively large number of features compared to single-layer approaches. Future research will focus on developing methods to further minimize the number of selected features.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. NRF-RS-2025-24683865).

REFERENCES

[1] Qwen, "Qwen2.5 technical report," arXiv preprint, arXiv:2412.15115, Dec 2024.

[2] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," arXiv preprint, arXiv:2211.09527, Nov 2022.

[3] Y. Liu et al., "Prompt injection attack against llm-integrated applications," arXiv preprint, arXiv:2306.05499, Jun 2023.

[4] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection," Proc. 16th ACM Workshop Artif. Intell. Secur., pp. 79–90, Nov 2023.

[5] M. Sawtell, T. Masterman, S. Besen, and J. Brown, "Lightweight safety classification using pruned language models," arXiv preprint, arXiv:2412.13435, Dec 2024.

[6] R. K. Sharma, V. Gupta, and D. Grossman, "SPML: A DSL for defending language models against prompt attacks," arXiv preprint, arXiv:2402.11755, Feb 2024.

[7] L. Zheng et al., "LMSYS-Chat-1M: A large-scale real-world LLM conversation dataset," International Conference on Representation Learning, 2024.

[8] L. Li et al., "SALAD-Bench: A hierarchical and comprehensive safety benchmark for large language models," Findings of the Association for Computational Linguistics: ACL 2024, pp. 3923–3954, Aug 2024.