
SENTINEL: SOTA MODEL TO PROTECT AGAINST PROMPT INJECTIONS

Dror Ivry
Qualifire
Tel Aviv, IL
dror@qualifire.ai

Oran Nahum
Qualifire
Tel Aviv, IL
oran.nahum@qualifire.ai

ABSTRACT

Large Language Models (LLMs) are increasingly powerful but remain vulnerable to prompt injection attacks, where malicious inputs cause the model to deviate from its intended instructions. This paper introduces Sentinel, a novel detection model, `qualifire/prompt-injection-sentinel`, based on the `answerdotai/ModernBERT-large` architecture. By leveraging ModernBERT's advanced features and fine-tuning on an extensive and diverse dataset comprising a few open-source and private collections, Sentinel achieves state-of-the-art performance. This dataset amalgamates varied attack types, from role-playing and instruction hijacking to attempts to generate biased content, alongside a broad spectrum of benign instructions, with private datasets specifically targeting nuanced error correction and real-world misclassifications. On a comprehensive, unseen internal test set, Sentinel demonstrates an average accuracy of 0.987 and an F1-score of 0.980. Furthermore, when evaluated on public benchmarks, it consistently outperforms strong baselines like `protectai/deberta-v3-base-prompt-injection-v2`. This work details Sentinel's architecture, its meticulous dataset curation, its training methodology, and a thorough evaluation, highlighting its superior detection capabilities.

Keywords Prompt Injection · Large Language Models · LLM Security · ModernBERT · Jailbreak

1 Introduction

Large Language Models (LLMs) have revolutionized numerous domains, showcasing remarkable capabilities in understanding, generation, and agentic applications. However, a critical vulnerability inherent in many LLMs is their difficulty in distinguishing between trusted system instructions and untrusted user-provided data. This vulnerability is exploited by prompt injection attacks, where malicious inputs manipulate the model into ignoring its original instructions and executing unintended, often harmful, actions [7, 14]. The fundamental issue lies in the LLM’s instruction-following ability being co-opted.

Existing detection methods for prompt injection face several limitations. Trained detectors, while sometimes effective, can exhibit biases towards their training data, leading to poor generalization on unseen attack vectors. For instance, the `protectai/deberta-v3-base-prompt-injection` model, while demonstrating impressive results on some benchmarks, showed significantly lower performance when evaluated on a more diverse dataset, suggesting a potential training data bias. The dynamic nature of attacks, often described as a "cat-and-mouse game", means that defenses trained on known attack signatures can quickly become outdated.

This paper introduces Sentinel, a robust detection model officially named `qualifire/prompt-injection-sentinel` [18]. It is built by Qualifire upon the `answerdotai/ModernBERT-large` architecture [4, 23]. Sentinel is specifically trained to classify prompts as either benign or indicative of a jailbreak/injection attempt. By leveraging a meticulously aggregated and diverse dataset, Sentinel demonstrates superior performance. The primary contributions of this work are: (1) The development and comprehensive evaluation of Sentinel. (2) The curation of a diverse training and testing dataset. (3) A comparative analysis showcasing Sentinel’s significant improvements over established baselines on both a comprehensive internal test set and public benchmarks.

2 Related Work

2.1 Prompt Injection Attacks

Prompt injection attacks aim to subvert an LLM’s intended purpose by embedding malicious instructions within the input prompt [7, 14]. These attacks can range from simple commands like "Ignore previous instructions" to more sophisticated techniques involving role-playing, obfuscation, character escaping, or context ignoring [1]. Recent research highlights various attack patterns, including direct injection, indirect prompt injection, and jailbreaks designed to bypass safety alignments [1, 3, 24]. The continuous evolution of attack methods, including template-based, generative, and optimization-driven approaches [3], necessitates robust detection mechanisms.

2.2 Prompt Injection Defenses and Detection

A primary approach to mitigation is the development of trained detector models. These are typically binary classifiers fine-tuned on datasets containing examples of both benign and malicious prompts. Notable examples include models based on architectures like DeBERTa [8], such as `protectai/deberta-v3-base-prompt-injection` and its subsequent versions [15, 16]. Other defense strategies include adversarial training and input preprocessing [24]. The effectiveness of trained detectors heavily depends on the quality and diversity of their training data. Recent studies emphasize the importance of dynamic benchmarks for evaluating defenses, as even strong cutting edge models can exhibit vulnerabilities [3, 24].

3 The Sentinel Detection Model

3.1 Model Architecture: Leveraging ModernBERT

Sentinel utilizes `answerdotai/ModernBERT-large` as its base. ModernBERT is a modernized bidirectional encoder-only Transformer model pretrained on 2 trillion tokens of English and code data, with a native context length of up to 8,192 tokens [23]. The "large" variant consists of 28 layers and 395 million parameters. Key architectural features include:

- **Rotary Positional Embeddings (RoPE):** [21] For superior long-context support and relative position encoding.
- **Local-Global Alternating Attention:** For efficient processing of long input sequences.
- **Unpadding and Flash Attention:** [5] For optimized inference speed and memory efficiency.

3.2 Dataset Curation and Preparation

A key component of Sentinel’s development was the creation of an extensive and diverse dataset for training and evaluation.

3.2.1 Open-Source Datasets

The following open-source datasets were incorporated:

- Salad-Data [13]: Filtered for ‘O5: Malicious Use’ category, this dataset is known for including creative and complex jailbreak attempts.
- alespalla/chatbot-instruction-prompts [2]: A source of benign prompts (7,000 samples used).
- microsoft/orca-agentinstruct-1M-v1 [12]: Benign prompts (7,000 samples extracted from the "content" field).
- verazuo/jailbreak-llms [20]: A collection of jailbreak and benign prompts from an in-the-wild repository.
- lmsys/toxic-chat [11]: Used for jailbreak prompts where the "jailbreaking" column indicated an attack.
- VMware/open-instruct [22]: A source of benign prompts (7,000 samples used).
- reshahs/SPML-Chatbot-Prompt-Injection [19]: Contains 16,000 samples focusing on scenario-based attacks.

3.2.2 Private Datasets

To address specific challenges, Qualifire developed a private dataset:

- qualifire-synthetics: Contains 1,400 synthesized using LLMs.

3.2.3 Final Dataset Composition and Splitting

After consolidating all sources, the dataset was structured to have approximately 70% benign and 30% jailbreak prompts. The entire dataset was then split into a 90% training set and a 10% test set, ensuring no overlap between them.

3.3 Training

Sentinel was developed by fine-tuning the `answerdotai/ModernBERT-large` model on the 90% training split of the curated dataset. The task was formulated as a binary classification problem.

4 Experimental Setup

4.1 Test Dataset

Evaluation was conducted on two fronts: (1) the 10% held-out internal test set, comprising a diverse mix of prompts from all source datasets, and (2) several public, standardized prompt injection benchmarks.

4.2 Baseline Model

The primary baseline for comparison is `protectai/deberta-v3-base-prompt-injection-v2` [16].

4.3 Evaluation Metrics

For the internal test set, we used Accuracy (AvgAcc), Recall, Precision, and F1-score. For public benchmarks, we report the Binary F1 Score as is standard.

5 Results and Analysis

5.1 Performance on Internal Test Set

On our comprehensive internal test set, Sentinel demonstrated significantly superior performance compared to the baseline. The results are summarized in Table 1.

Table 1: Performance on the Internal Held-Out Test Set

Model	AvgAcc	Recall	F1	Precision	Params
qualifire/prompt-injection-sentinel [18]	0.987	0.991	0.980	0.986	0.395B
protectai/deberta-v3-base-prompt-injection-v2 [16]	0.848	0.905	0.728	0.820	0.185B

Sentinel achieved an average accuracy of 0.987, surpassing the baseline by 13.9 percentage points. The F1-score of 0.980 is substantially higher than the baseline’s 0.728. The superior performance is attributed to both the advanced ModernBERT architecture and the extensive and diverse training dataset.

5.2 Benchmark Performance

To further validate Sentinel’s robustness and generalization, we evaluated it on four challenging public prompt injection benchmarks. As shown in Table 2, the Sentinel model consistently and significantly outperforms the strong DeBERTa-v3 baseline across all datasets.

Table 2: Benchmark Performance (Binary F1 Score)

Model	allenai/wildjailbreak [10]	jackh hao/jailbreak -classification [9]	deepset/prompt -injections [6]	qualifire/Qualifire-prompt -injection-benchmark [17]	Avg
qualifire/prompt-injection-sentinel [18]	0.935	0.985	0.857	0.976	0.938
protectai/deberta-v3-base-prompt-injection-v2 [16]	0.733	0.915	0.536	0.652	0.709

The results on these public benchmarks confirm the findings from our internal test set. Sentinel’s average F1 score of 0.938 is nearly 23 points higher than the baseline’s 0.709. This consistent out performance provides strong evidence of the model’s superior generalization capabilities.

5.3 Latency and Hardware

This benchmark was performed using an L4 GPU. Due to the extremely small size of the model the latency was quite surprising at an avg latency of just ~0.02 seconds per inference time we achieved incredible real-time evaluation capabilities on a fairly modest hardware.

6 Discussion and Limitations

The results indicate that Sentinel sets a new standard for open-source prompt injection detection. The improvement over the baseline highlights the benefits of using advanced base models and investing in extensive dataset curation.

Despite its strong performance, Sentinel has limitations:

1. **Susceptibility to Novel Attacks:** As a trained model, its knowledge is based on its training data. Highly novel attack vectors could potentially evade detection [24].
2. **Dataset Reproducibility:** The inclusion of private datasets means that exact replication of the training environment is contingent on access to these proprietary assets.

6.1 Error analysis

To better understand Sentinel’s limitations, we conducted a manual review of a random sample of misclassifications from the internal test set. Surprisingly, the errors did not fall into distinct or recurring categories. Instead, we didn’t observed any specific identifiers or characteristics for the errors observed. False positives (benign prompts incorrectly classified as injections) - typically included edge cases involving unusual formatting, assertive or security-related

phrasing, or ambiguous intent. False negatives (missed jailbreaks) - tended to involve subtle adversarial phrasing that did not strongly resemble known attack patterns.

7 Conclusion and Future Work

Sentinel has demonstrated state-of-the-art performance in prompt injection detection. This success underscores the importance of advanced model architectures and comprehensive training data.

Future work will focus on:

- **Continuous Dataset Evolution:** Regularly updating the training dataset with new jailbreak techniques.
- **Model Optimization:** Exploring techniques like knowledge distillation and quantization to create smaller, faster versions of Sentinel.
- **Hybrid Defense Approaches:** Investigating the integration of Sentinel with other defense mechanisms, such as input sanitization or runtime monitoring [3].

A How to Get Started with the Model

This section provides a simple code snippet to get started with the `qualifire/prompt-injection-sentinel` model using the Hugging Face `transformers` library. Ensure you have the library installed (`pip install transformers torch`).

A.1 Code Example

```

1 from transformers import pipeline, AutoTokenizer,
   AutoModelForSequenceClassification
2
3 model_id = 'qualifire/prompt-injection-sentinel'
4
5 # Load the tokenizer and model from Hugging Face Hub
6 tokenizer = AutoTokenizer.from_pretrained(model_id)
7 model = AutoModelForSequenceClassification.from_pretrained(model_id)
8
9 # Create a text-classification pipeline
10 pipe = pipeline("text-classification", model=model, tokenizer=tokenizer)
11
12 # Test with a benign prompt
13 result = pipe("hi how are you?")
14
15 print(result)

```

Listing 1: Python code to run the Sentinel model.

A.2 Example Output

The code above will produce the following output for a benign prompt, indicating a high confidence score for the 'benign' label.

```
1 [{"label": "benign", "score": 1.0}]
```

References

- [1] Abdalrahman Al-Kaswan, Zhaohan Yao, Sijia Liu, and Pin-Yu Chen. Is your prompt safe? investigating prompt injection attacks against open-source llms. *arXiv preprint arXiv:2505.14368*, 2025.
- [2] alespalla. alespalla/chatbot_instruction_prompts dataset. https://huggingface.co/datasets/alespalla/chatbot_instruction_prompts, 2023. Accessed: June 1, 2025.
- [3] Anonymous. Evolving security in llms: A study of jailbreak attacks and defenses. *arXiv preprint arXiv:2504.02080*, 2025.

- [4] Answer.AI. answerdotai/ModernBERT-large model card. <https://huggingface.co/answerdotai/ModernBERT-large>, 2024. Accessed: June 1, 2025.
- [5] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16284–16299, 2022.
- [6] deepset. deepset/prompt-injections dataset. <https://huggingface.co/datasets/deepset/prompt-injections>, 2023. Accessed: June 1, 2025.
- [7] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you’ve signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 25–36, 2023.
- [8] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*, 2021.
- [9] jackhhao. jackhhao/jailbreak-classification dataset. <https://huggingface.co/datasets/jackhhao/jailbreak>, 2023. Accessed: June 1, 2025.
- [10] Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *arXiv preprint arXiv:2406.18510*, 2024.
- [11] lmsys. lmsys/toxic-chat dataset. <https://huggingface.co/datasets/lmsys/toxic-chat>, 2023. Accessed: June 1, 2025.
- [12] Microsoft. microsoft/orca-agent-instruct-1M-v1 dataset. <https://huggingface.co/datasets/microsoft/orca-agent-instruct-1M-v1>, 2024. Accessed: June 1, 2025.
- [13] OpenSafetyLab. OpenSafetyLab/Salad-Data dataset. <https://huggingface.co/datasets/OpenSafetyLab/Salad-Data>, 2024. Accessed: June 1, 2025.
- [14] F. Perez and I. Ribeiro. Ignore previous prompt: An empirical analysis of characterizing and mitigating prompt injection attacks. *arXiv preprint arXiv:2211.09527*, 2022.
- [15] ProtectAI. protectai/deberta-v3-base-prompt-injection model card. <https://huggingface.co/protectai/deberta-v3-base-prompt-injection>, 2023. Accessed: June 1, 2025.
- [16] ProtectAI. protectai/deberta-v3-base-prompt-injection-v2 model card. <https://huggingface.co/protectai/deberta-v3-base-prompt-injection-v2>, 2023. Accessed: June 1, 2025.
- [17] Qualifire. qualifire/Qualifire-prompt-injection-benchmark Dataset. <https://huggingface.co/datasets/qualifire/Qualifire-prompt-injection-benchmark>, 2025. Accessed: June 1, 2025. Please update URL if it changes.
- [18] Qualifire. qualifire/prompt-injection-sentinel Model Card. <https://huggingface.co/qualifire/prompt-injection-sentinel>, 2025. Accessed: June 1, 2025. Please update URL if it changes.
- [19] reshabhs. reshabhs/SPML-Chatbot-Prompt-Injection dataset. <https://huggingface.co/datasets/reshabhs/SPML-Chatbot-Prompt-Injection>, 2024. Accessed: June 1, 2025.
- [20] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. “Do Anything Now”: Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024.
- [21] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunbo Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.
- [22] VMware AI Labs. VMware/open-instruct dataset. <https://huggingface.co/datasets/VMware/open-instruct>, 2023. Accessed: June 1, 2025.
- [23] Brennan Warner, Alex Chaffin, Benjamin Clavié, Orion Weller, Oliver Hallström, Salma Taghadouini, Andrew Gallagher, Ritam Biswas, Feroze Ladhak, Thijs Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*, 2024.
- [24] Yaguan Zhang, Jia Li, Zhaofeng Wang, Xinyi Chen, Yi Liu, and Lin Song. OET: An optimization-based evaluation toolkit for benchmarking prompt injection attacks and defenses. *arXiv preprint arXiv:2505.00843*, 2025.