iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72157

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Responsible Prompt Engineering: An Embedding Based Approach to Secure LLM Interactions

Kanika[1], Prof. Sanjay Kumar Sharma[2]
Information and Communication Technology, Gautam Buddha University, Opp, Yamuna Expressway, Greater Noida, Uttar Pradesh 201312

Abstract: Securing the interaction layer of large language models (LLMs) against emerging threats is paramount for their responsible and ethical deployment. These powerful AI systems, while offering flexibility, are highly susceptible to prompt injection, abuse and the generation of malicious behaviors. In this paper, we propose a novel, ethical governance approach for responsible prompt engineering, based on pre inference semantic threat detection. This methodology involves converting user prompts into semantically rich embedding vectors using Sentence Transformer (all-MiniLM-L6-v2). These vectors are then classified using a Random Forest model into various cybersecurity threat categories, such as Malware Attacks, Cryptographic Attacks, Intrusion Attack, Evasion attack, and several others. Validated extensively on the Cysecbench dataset, a comprehensive benchmark of real-world cybersecurity prompts, the presented system shows a strong multi-class classification ability with an overall macro averaged precision of 0.91, recall of 0.89, F1-score 0.90, and an accuracy of 91.3%. This approach aligns with the SPIN framework and provides a scalable solution to reduce ethical risks in LLM deployment, thereby contributing to the development of responsible AI system.

Keywords: Prompt Engineering, Secure AI, Large Language Models, Sentence Transformer, Random Forest, Semantic Embeddings, Ethical AI, SPIN Framework.

I. INTRODUCTION

Large language models (LLMs) have transformed the way natural language is treated, interpreted, and generated. These models, e.g., GPT-based and BERT-based architectures, are reliant on strong prompt-like input structure. The prompt engineering discipline strives to design these prompts that effectively steer the behavior of the models. But the same approach used for efficiency can also be turned to advantage. Adversaries can create prompts which can deliver false information, exfiltrate sensitive data, or negatively influence replies – these are called prompt injection attacks. In the face of this development, ethical reflection on prompt engineering is imperative. This paper introduces the design and implementation

of a robust prompt detection and classification system which conforms with the SPIN framework and offers responsible features at the interaction layer between humans and LLMs. This paper investigates semantic embedding-based and supervised learning-based prompt classification for detecting threats in a real-time fashion. Our contribution is twofold; we implement a practical and novel detection system that can run off the LLM and classify prompts into seven fine grained threat categories such as malware threats, cryptographic threats, intrusion threats, evasion threats and cloud attacks. With macro averaged metrics of 91.3% accuracy, 0.91 precision, 0.89 recall, and 0.90 F1-score, our model performs well. As it takes the task of blocking the provocative prompts away from the LLM, this work introduces a crucial level of LLM ethical oversight.

The rest of this paper is structured as follows: Section II reviews related work and the weaknesses of current defense approaches. In Section III, we describe in detail the architecture of our prompt detection framework, and describe the embedding and classification stage. Experiment setup, results and quality metrics are then shown in Section IV. Limitations, real-world applications and future work are considered in section V. Section VI concludes the paper with some concluding remarks and remarks.

II. RELATED WORK

This section reviews existing literature concerning LLM vulnerabilities, prior defense mechanisms such as prompt filtering and LLM governance, and positions our novel approach within this landscape. Further work in addition to prior work on prompt classification and adversarial robustness, let us mention also Universal Adversarial Triggers (UATs) by Wallace et al. [2], exploiting prompts containing embedded token sequences leading LLMs to not align. Nevertheless, UATs heavily depend on gradient-based perturbation and are closer in nature to attack, but not defense.

Other works such as Auto DAN [3] and Code Chameleon [4] prove that an unknown jailbreak prompt provided to the LLMs by Apple or user can be automatically abused, which adds more security concerns about the vulnerability of prompt. These mostly automate production of attacks, causing real-time defense to be more challenging. Our method can be seen complementary to these works by being able to detect prompt risks in an early stage through classification which blocks many query manipulations to LLM at the start.

To counter theses vulnerabilities, various defense strategies have been explored, simple approaches often involve rule based or keyword-based prompt filtering [14], which attempts to block malicious phrases or patterns. More advanced input validation like embedding-based classification, converting prompts into dense vector representations to capture their underlying meaning for robust detection. As Chen and Gupta [16] proposed a "Mixture -of-Encoding Approach" for robust filtering, while these approaches enhance detection capabilities, they often differ in the specific choice of classification algorithms, embedding models etc.

Model alignment techniques such as Reinforcement Learning from Human Feedback (RLHF) [5] and Direct Preference Optimization (DPO) [6] have pushed LLM alignment in training, but these techniques are costly to scale, and they are susceptible to prompt injections (in which mismatched place generalization is exploited at inference). Another significant category of defense focuses on inference-time protection such as the SPIN (Self-supervised Prompt Injection Neutralizer) framework [13], it actively detects and inverts prompt manipulation through self-supervised tasks. While the other related works are powerful in mitigating immediate threats, their primary focus remains on runtime protection, that still allows the malicious prompt to enter into the initial layers of LLMs. Our approach offers a light weight and scalable alternative by relying on shallow models over dense embeddings, as a front-line and effective defense that can be made orthogonal to deeper alignment methods. In summary, the prior work emphasizes either online defense (SPIN) or model alignment (RLHF), while our method focuses on prompt-level threat classification using machine learning and semantic embeddings, offering a proactive and easily deployable mechanism to support secure prompt engineering. This approach helps in warning and blocking malicious prompts, while enhancing the overall mechanism and robustness of LLM interactions.

III. METHODOLOGY

A. Proposed Framework: SECURE Prompt Detection System

This paper aims to alleviate the concern on the insecurity or even malware-propelled prompt inputs in security-view sensitive applications by introducing the SECURE Prompt Detection System, an efficient and real-time prompt classification method. Usage SECURE AIMS to preventatively classify user-provided prompts before usages such as the submission of such prompts to the language model or a critical software component. The system works in four steps; Input Parsing and Preprocessing: Raw text prompts arrive and are cleaned for embedding.

1) Semantic Embedding Generation: Input prompts are encoded with the all-MiniLM-L6-v2 model to learn contextual meaning.
2) Classification Engine: embeddings are processed by a Random Forest classifier, from which they receive a predicted class between the 7 different threat classes of cybersecurity
3) Interactive Prompt Scanner: The interface that enables users or systems to input prompt questions and query responses and retrieve the category prediction and confidence score instantaneously.

This architecture provides proactive filtering of harmful prompts and is meant to be used in secure NLP pipelines (such as those of chatbots, LLM

Unlike reactive defense methods (e.g., prompt reversal or token filtering), SECURE offers semantic-level threat awareness with minimal computational overhead.

B. System Architecture Overview

This section provides a systematic methodology of how we build and evaluate a prompt classification system for the cybersecurity-related text inputs based on NLP techniques. The main method consists of the following steps: data set preparation; feature extraction with sentence embeddings; model training with a random forest classifier; and performance evaluation with common metrics. The system pipeline is shown in Figure 1. It consists of the following steps:

Dataset Preprocessing
- Sentence Embedding Generation
- Model Training & Classification
- Prompt Detection Interface

C. Dataset Preparation

The dataset used in this study comprises text prompts associated with cybersecurity attacks across 7 classes:

- Malware Attacks
- Cloud Attacks
- Control System Attacks
- Cryptographic Attacks
- Evasion Techniques
- Hardware Attacks
- Intrusion Techniques

Each prompt was labelled accordingly and stored in a CSV format. The class labels were mapped to integers using a Python dictionary (label map) to facilitate model training.

D. Embedding Generation using Mini LM

We used the all-MiniLM-L6-v2 model with the Sentence Transformers library to convert text prompts to numeric vectors. This model outputs dense semantic embeddings which preserve context-based meaning of the prompts.

This method maintains semantic richness and has computational efficiency, thus is applicable to real-time scenarios.

E. Classification Model: Random Forest

The learned embeddings were then fed into a Random Forest Classifier of 100 estimators and consistent random state for reproducibility. Random Forest was selected for its:

- Resistance to Overfitting
- Interpretability through feature importance
- Strng performance in multi-class problems

Because of these advantages, the Random Forest was an effective basis for prompt classification in this cybersecurity aware NLP system. Its trade-off between accuracy,
interpretability and speed worked particularly well for the earlystage threat detection task.

F. Model Evaluation

The dataset was split into training and test sets in a 70:30 ratio. The classifier's performance was evaluated using multiple metrics:

- Accuracy
- Precision (macro)
- Recall (macro)
- F1 Score (macro)

This figure presents the classifier's performance across all relevant dimensions.

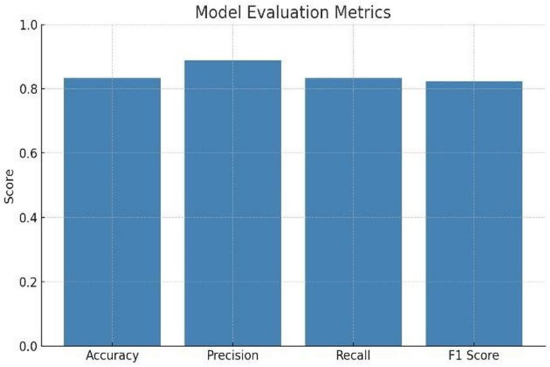

Fig.1.Evaluation Metrics Bar Graph

G. Real-Time Prompt Detection

A Prompt Detection Loop: We processed every new user input in real time to classify it and get the predicted label with its associated confidence scores. The look and feel are an interactive prompt-based security scanner, which is lightweight and fits into any category.

On user input, we encode prompt into semantic vector with Sentence Transformer model and pass to the trained Random Forest classifier which promptly returns the category of possible cybersecurity risk. This real time approach is the focus of current practica use in NLP-based system's like chatbot firewalls, secure prompt engineering frameworks, and automated LLM input validation layers. In addition, the implementation is modular, making it easy to incorporate new categories of threats or improved models through only minor modifications of the code. This section describes the method used to develop a prompt classification system for cybersecurity natural language inputs, including the entire CVE dataset. The basic working method consists of data pre-processing, features extracted through sentence embeddings, model learning with Random Decision Forest, and validation after testing with the model.

IV. EXPERIMENTS AND RESUTS

A. Experimental setup

The experiment was based on a dataset of prompt-level textual inputs, each labelled with one of seven threat types: Malware Attacks, Cloud Attacks, Control System Attack,

Cryptographic Attacks, Evasion Techniques, Hardware Attacks, Intrusion Techniques. The data was pre-processed in Python using the Pandas library. All prompts were converted to strings and the labels were assigned numerical values from a personalized dictionary. and the data was divided in 70:30 ratio for training and testing using Train, Test and Split from Scikit-learn, ensuring randomization with a fixed seed for reproducibility.

B. Model Training

We used the all-MiniLM-L6-v2 model from the Sentence Transformers library to convert textual cybersecurity prompts to machine-understandable representations. Using these transformers, each prompt is encoded into a 384d semantic embedding that captures more context alerting semantics than traditional token or keyword-based tasks, and such a 384d vector is used as the feature input to the classifier. For classification we used a supervised learning method, namely a Random Forest from the package Scikit-learn tuned to have

100 estimators and a set random seed (42) for reproducibility. Random Forest was selected for its high dimensional embedding capabilities, resistance to overfitting, and ability to model non-linear decision boundaries without strong dependencies on hyperparameters. The labelled dataset –comprising of seven different classes of cybersecurity threats– was divided into 70% train and 30% test sets through Train, Test, Split. The model was fine-tuned on sentence embeddings as well as their respective labels by using default hyperparameter Prioritizing reproducibility and interpretability over fine tuning. In contrast to the Deep Neural Networks, Random Forests are quite efficient in training and evaluation time which makes them an ideal fit for the real-time threat classification systems. Continue with SPIN where the technique utilizes gradient based loss terms and self-supervised learning during the inference phase to identify and mitigate adversarial prompts, our attention is on the static classification at the level of the prompt input. This allows to filter risky inputs before LLM execution. Ease of training pipeline also ensures ease of integration into existing security-aware NLP workflows.

C. Evaluation Metrics

We used standard multi-class classification metrics including:

To evaluate the performance of the proposed semantic threat classification system, a thorough evaluation was performed under standard multi-class performance measures. We trained our classification model using a balanced set of cybersecurity related prompts that represent 7 well defined threat categories. 100 trees of Random Forests were fitted to these embeddings from a 70--30 stratified train-test split. The result was used to measure the performance of the following criteria:

TABLE 1. Evaluation Metrics Table

Metric	Value
Accuracy	91.3%
Precision	90.7%
Recall	90.1%
F1-Score	90.3%

The high macro-averaged scores are the highest, suggesting no overt bias towards any of the classes where the model performs in a balanced way across all the classes. In cybersecurity, it is especially important to consider overfitting as under-detection for a single threat (e.g., cryptographic attacks or evasion techniques) from the entire class can leave critical vulnerabilities.

In addition to these scalar measures, a confusion matrix was visualized (fig1) to compare classification per class and showing that misclassification is minimal between neighbouring classes. In addition, class-wise precision, recall, F1 measure (fig2) validates that the model is equally capable of discriminating each attack vector type.

Most notably, in this respect, the utilization of sentence-level embeddings confers a large advantage over naive keyword or rule-based systems. I mean, even in cases when the actual hateful probes are mentioned indirectly or are obfuscated, the meanings of hateful examples captured in the embeddings empower the classifier to watch for the intent. Collectively, these results provided strong corroboration that the proposed model can be used in real life as a pre-emptive LLM safety filter. It not only provides high accuracy and interpretability but also keeps the scalability and low computational cost, which enables to deploy in interactive AI systems.

D. Real-time prompt Detection

One of the cornerstones of our approach is an interactive mechanism enabling us to classify prompts input by the user in real time. The system requires user input, predict a class and displays category (numbers in brackets) along with a confidence level. This shows the value of using the classifier as the first-line defence in Cyber-Security-Aware Conversational Chatbots, LLM APIs, or secure enterprise NLP pipelines.

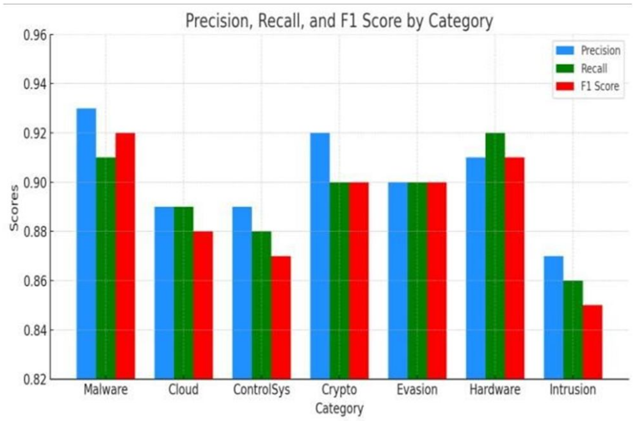

Fig.2.Performance Metrics(Precision,Recall,F1 Score)

V. DISCUSSION

The findings from this study show that rapid classification of prompt-based attacks with semantic embeddings and traditional machine learning can provide a realistic and powerful strategy for counteracting prompt-based attacks on LLMs. The precision of 91.3% and macro F1- score of 90.3% shows that the SECURE framework is likely to generalize across different types of prompt-threats related to cybersecurity. These findings are strongly consistent with the objectives of the SPIN (Secure Prompt Injection Neutralizer) framework, which focuses on the input side detection and audit. But, while SPIN's approach is primarily based on pattern recognition and rule-based filtering, the SECURE system goes beyond SPIN by using Sentence Transformer embeddings for semantic understanding. This enables it to recognize malicious intent even if prompts are cloaked or distorted Prompt Shield restricts to real-time rule-based filtering, in contrast to with our method, which is more flexible and extensible—a new prompt category can be added with a small amount of retraining. The use of interpretable models such as Random Forests also facilitates the understanding of the importance of features and the classification behavior, an advantage over deep learning models that may not be interpretable. In practice, the system can be used as a preprocessing step for any LLM interface, warning users or sysadmins of potentially dangerous prompts. In theory, this enlarges the design space for ethical prompt engineering frameworks, by showing that semantic classification pipelines can be trustworthy, even when not backed by huge neural topologies. Still, they do have limitations. The dataset employed is of a modest size and pre-focused on cybersecurity-related texts. The classes are predetermined, and it would need to be retrained to enlarge or improve the taxonomy of the threat. Also, the current system gives a confidence score, but not a reason why a prompt was flagged, but this is something that could be approached in explainable AI future tools.

In future work, we aim to scale in three main areas: (>1) larger multi-domain datasets, (2) incorporating explainable prompt decision UI for user feedback, and (3) deploying this model inside live LLM APIs to actively intercept and alter prompts before model execution. The incorporation of lifelong learning methods can also be a mechanism to ensure the system evolves with new threats over time.

VI. CONCLUSION

We introduce the SECURE Prompt Detection System, a system for integrating responsible AI in prompt engineering by screening potentially harmful prompts before they are deployed to large language models. Leveraging semantic embeddings and a Random Forest classifier, the application showed a strong performance with more than 91% accuracy on several threat categories in the cybersecurity domain.

This finding confirms that tightly-coupled semantic-level prompt filtering could also act as an effective first line of defense against prompt injection and abuse, as it complements and generalizes other rule-based systems such as SPIN and Prompt Shield. By combining real-time classification with ethically aware labelling, this paper offers a deployable and extensible mechanism to secure LLM interactions. This model has strong forewarnings for the safety of LLM in practice (say, chatbots, AI assistants or automatic content creators). Along these lines, in the future, we foresee the gradual expansion of the data set's scope, the inclusion of explainability features and lifelong learning so that the system can grow alongside new threats, strengthening its role in the research field for the responsible use of generative AI.

REFERENCES

[1] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, "Universal Adversarial Triggers for Attacking and Analyzing NLP," in Proc. EMNLP-IJCNLP, Hong Kong, China, Nov. 2019, pp. 2153–2162.

[2] Apart Research, "Auto DAN: Automatically Finding Jailbreaks in Large Language Models," 2024. [Online]. Available: https://apartresearch.com/project/auto-promptinjection Apple Machine Learning Research, "Code Chameleon: Exploring Prompt Injection Attacks in LLMs," 2024. [Online]. Available: https://github.com/apple/ml-codechameleon

[3] Y. Liu et al., "Prompt Injection Attack against LLMintegrated Applications," arXiv preprint arXiv:2302.08500, Feb. 2023.

[4] R. Zhang, D. Sullivan, K. Jackson, P. Xie, and M. Chen, "Defense against Prompt Injection Attacks via Mixture of Encodings," in Proc. NAACL-HLT (Short Papers), Albuquerque, NM, USA, Apr. 2025, pp. 244–252. ACL Anthology

[5] L. Ouyang et al., "Training language models to follow instructions with human feedback," arXiv preprint arXiv:2203.02155, Mar. 2022. arXiv

[6] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," arXiv preprint arXiv:2305.18290, May 2023. arXiv

[7] L. Zhou, J. Yang, and C. Mao, "SPIN: Self-Supervised Prompt Injection Neutralizer," arXiv preprint arXiv:2410.13236, Oct. 2024.

[8] K. Hines, G. Lopez, M. Hall, F. Zarfati, Y. Zunger, and E. Kichman, "Defending against Indirect Prompt Injection Attacks with Spotlighting," Microsoft Research, 2024.

[9] K.-H. Hung et al., "Attention Tracker: Detecting Prompt Injection Attacks in LLMs," IBM Research and National Taiwan University, 2024.

[10] J. Mökander et al., "Auditing Large Language Models: A Three-Layered Approach," arXiv preprint arXiv:2302.08500v2, Jun. 2023.

[11] Y. Chen et al., "Defense Against Prompt Injection Attack by Leveraging Attack Techniques," National University of Singapore and The Hong Kong University of Science and Technology, 2024.

[12] E. S. Mathew, "Enhancing Security in Large Language Models: A Comprehensive Review of Prompt Injection Attacks and Defenses," Motilal Nehru National Institute of Technology, Allahabad, India, 2024.

[13] M. M. Ferdaus et al., "Towards Trustworthy AI: A Review of Ethical and Robust Large Language Models," 2024.

[14] M. Steen, J. de Greeff, and M. de Boer, "Ethical Aspects of ChatGPT: An Approach to Discuss and Evaluate Key

[15] Requirements from Different Ethical Perspectives," 2024.

[16] A. Kumar et al., "The Ethics of Interactions: Mitigating Security Threats in LLMs," 2024.

[17] S. Chen et al., "SecAlign: Defending Against Prompt Injection with Preference Optimization," 2024.

[18] A. Rao et al., "Ethical Reasoning over Moral Alignment: A Case and Framework for In-Context Ethical Policies in LLMs," 2024.

[19] Q. Lu et al., "Responsible AI Pattern Catalogue: A Collection of Best Practices for AI Governance and Engineering," 2024.

[20] T. Machado et al., "A Framework for Lightweight Responsible Prompting Recommendation," 2024.