ಬಿ.ಎಂ.ಎಸ್. ತಾಂತ್ರಿಕ ಮತ್ತು ವ್ಯವಸ್ಥಾಪನಾ ಮಹಾವಿದ್ಯಾಲಯ
BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
(Autonomous Institution Under VTU)
Yelahanka, Bengaluru -560119

A Major-project Phase 1 Work Report

on

"Adaptive Context-Aware Multi-Layer Prompt Security and Output Governance Framework for Secure Large Language Model Deployment"

Submitted in partial fulfillment for the award of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE AND ENGINEERING

by

Mr. Shakthi N V	1BY23IS194
Ms. Sowrabhi Narayanan	1BY23IS214
Mr. Vineeth Sagar H L	1BY23IS250
Mr. M Sai Vikram	1BY23IS262

Under the Guidance of

Prof. Sonnegowda K
Assistant Professor

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
Avalahalli, Yelahanka, Bengaluru, Karnataka – 560119

April – 2026

ಬಿ.ಎಂ.ಎಸ್. ತಾಂತ್ರಿಕ ಮತ್ತು ವ್ಯವಸ್ಥಾಪನಾ ಮಹಾವಿದ್ಯಾಲಯ
BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
(Autonomous Institution Under VTU)
Yelahanka, Bengaluru -560119

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major-Project work (BCS606) entitled "**Adaptive Context-Aware Multi-Layer Prompt Security and Output Governance Framework for Secure Large Language Model Deployment**" is a bonafide work carried out by **Mr. Shakthi N V(1BY23IS194), Ms. Sowrabhi Narayanan (1BY23IS214), Mr. Vineeth Sagar H L(1BY23IS250) , Mr. M Sai Vikram(1BY23IS262)** in partial fulfillment for the award of **Bachelor of Engineering Degree in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2025–26. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The Major -project phase 1 report has been approved as it satisfies the academic requirements with respect to Major-project work for the B.E Degree.

Signature of the Guide (Prof. Sonnegowda K)	Signature of the Coordinator (Dr. Kalai Vani YS)	Signature of the Division Head 5 (Dr. N. Rakesh)	Signature of the HOD, ISE (Dr. Surekha K B)

ABSTRACT

Context: Large Language Models (LLMs) are increasingly deployed in enterprise software, systems and smart platforms. But with their increasing use, they've also raised significant vulnerabilities like prompt injections, adversarial attacks and unintentional data leakage. This demonstrates the need for more effective safety measures

Problem Identified: Current safety mechanisms are based on basic keyword filtering or single-stage moderation, which look at prompts in isolation. These approaches do not detectattacks, multiple-step attacks and attacks in multimodal features such as images or audio.

Objective: The aim of this project is to create and implement an intelligent, contextual multi-layered security system that can identify, block and control prompt-based attacks and facilitate secure and trustworthy interactions with LLM.

Methods: We employ a multi-layer prompt detection system that assesses text at lexical, semantic, contextual and behavioural levels. It also supports multi-modal processing of text, audio and images, and output control mechanisms like policy validation and sandbox verification.

Results: The system enhances the detection of indirect and multi-step attacks, and minimising by generating safe prompts and produces safe responses.It also provides explainable monitoring via decision logs.

Keywords: LLM Security, Prompt Injection, Multi-Layer Architecture, Context-Aware Analysis, Multi-Modal Processing, Output Governance, Explainable AI.

ACKNOWLEDGEMENT

We are happy to write a major-project report after completing it successfully. This major-project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude to each and everyone who has helped us to make this major project phase 1 a success.

We heartily thank **Dr. Sanjay H A**, Principal, BMS Institute of Technology Management for his constant encouragement and inspiration in taking up this major -project phase 1.

We heartily thank **Dr. Surekha K B**, Head of the Department, Information Science and Engineering, BMS Institute of Technology Management for constant encouragement and inspiration in taking up this major -project phase 1.

We heartily thank **Dr .N. Rakesh**, Associate Head of CSE Cluster -5 , Information Science and Engineering, BMS Institute of Technology Management for constant encouragement and inspiration in taking up this major -project phase 1.

We gratefully thank our mini-project guide, **Prof. Sonnegowda K**, Assistant Professor, Dept. of Information Science and Engineering, for his encouragement and advice throughout the course of the major -project phase 1.

We heartily thank major-project coordinator **Dr. Kalai Vani Y S** ,Assistant Professor, Dept. of Information Science and Engineering, for their constant follow-up and advice throughout the major -project phase 1.

Special thanks to all the staff and spark members of the Information Science and Engineering Department for their help and kind co-operation.

Lastly, we thank our parents and friends for their encouragement and support given to us in order to finish this major -project phase 1.

By,

Shakthi N V
Sowrabhi Narayanan
Vineeth Sagar H L
M Sai Vikram

ಬಿ.ಎಂ.ಎಸ್. ತಾಂತ್ರಿಕ ಮತ್ತು ವ್ಯವಸ್ಥಾಪನಾ ಮಹಾವಿದ್ಯಾಲಯ
BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
(Autonomous Institution Under VTU)
Yelahanka, Bengaluru -560119

Declaration

We, hereby declare that the major-project (BCS606) titled "**Adaptive Context-Aware Multi-Layer Prompt Security and Output Governance Framework for Secure Large Language Model Deployment**" is a record of original major-project work undertaken for partial fulfillment of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2025–26. We have completed this major-project work under the guidance of **Prof. Sonnegowda K, Assistant Professor**.

I also declare that this major-project report has not been submitted for the award of any degree, diploma, associateship, fellowship or other titles anywhere else.

Student Photo	Photo1	Photo2	Photo3	Photo4
				
Name	Shakthi N V	Sowrabhi Narayanan	Vineeth Sagar H L	M Sai Vikram
USN	1BY23IS194	1BY23IS214	1BY23IS250	1BY23IS262

Signature

Examiner's Signature

1.

2.

INDEX

ABSTRACT **3**

ACKNOWLEDGEMENT **4**

List of Figures **8**

1 INTRODUCTION **9**

1.1 Background 9

1.2 Security Challenges in LLM Systems 9

1.3 Limitations of Existing Systems 10

1.4 Proposed Approach 10

2 PROBLEM STATEMENT **11**

3 OBJECTIVES **12**

4 LITERATURE SURVEY **13**

4.1 Prompt Injection Attacks & Evaluation Frameworks 13

4.2 Prompt Injection Defense & Security Frameworks 14

4.3 Multimodal Prompt Injection & Vision-Language Attacks 14

4.4 Sophisticated & Multi-Agent Attack Strategies 15

4.5 Jailbreak Attacks through Audio & Speech 15

4.6 Context-Aware LLMs in Recommendation Systems 15

4.7 LLM-based Evaluation, Governance & Compliance 16

4.8 Secure & Private LLM Platforms 16

4.9 Cybersecurity & Adaptive Defense Systems 16

4.10 Survey of Jailbreak Attacks 17

5 RESEARCH GAP AND RESEARCH CHALLENGES **19**

6 PROPOSED METHODOLOGY **20**

6.1 Approach 20

6.2 System Architecture 21

6.2.1 Multi-Modal Input Layer 21
6.2.2 Text Normalization & Preprocessing Layer 21
6.2.3 Context Buffer Layer (Sliding Window) 21
6.2.4 Sliding Window Context Shift Analyzer (SWCSA) 21
6.2.5 Drift Score Calculation Layer 22
6.2.6 Intent Fragmentation & Reconstruction layer (IFS-R) 22
6.2.7 Policy and Safety Enforcement Layer 22
6.2.8 Target LLM Processing Layer 22
6.2.9 Supporting Components 23
6.3 Tools and Technologies 23
6.4 Process 24
6.4.1 Data Collection, Preprocessing and Multi-Modal Processing 24
6.4.2 Prompt Inspection and Context-Aware Analysis 25
6.4.3 Intent Reconstruction and Policy Enforcement 25
6.4.4 Post-Generation Governance and Monitoring 25
6.4.5 Evaluation Metrics 25
6.4.6 Technical and Economic Feasibility 26

7 HIGHLIGHTS FROM PHASE 1 REVIEW 1 **27**

8 EXPECTED OUTCOMES **28**

9 CONCLUSION **29**

10 REFERENCES **30**

LIST OF FIGURES

1 Context-Aware Prompt Processing and Security Pipeline 21

2 System Architecture of Adaptive Multi-Layer Prompt Security Framework 22

1 INTRODUCTION

1.1 Background

Large Language Models (LLMs) are a major area of growth within the Artificial Intelligence (AI) industry. Chatbots, content generation systems, and intuitive decision support systems are just some examples of the many applications for LLMs. As LLMs improve at reading and generating human-like text, they are increasingly being used in enterprise software, education, virtual assistants and automation workflows. LLMs' ability to quickly process natural language has made them a key feature of today's AI systems.

As LLMs continue to be more widely adopted, attackers have more opportunities to exploit LLMs using user-driven techniques. Attackers may use carefully crafted prompts to alter model output, disable safety features, or retrieve confidential data. Prompt injection attacks pose increasingly high risk when LLMs are connected to sensitive databases or enterprise systems, as unauthorized actions from an attacker using an LLM can introduce significant risks to information security and privacy.

1.2 Security Challenges in LLM Systems

Currently, LLMs are primarily protected through either static filtering or basic moderation techniques. Security for LLMs focuses on evaluating each prompt by itself without consideration for the previous prompt and is primarily based on keyword detection or some type of predefined "unsafe" instruction detection. Most modern attack vectors against LLMs are context-based, meaning that they develop over time and can be difficult to detect using only isolated prompt analysis.

The other challenge to protecting LLMs is that there is an increasing number of multi-modal inputs (e.g., text, audio, and image). Attackers can put malicious instructions in images or audio files, allowing them to bypass traditional text filtering. An additional type of attack on context-based LLM interactions is context poisoning. A context poisoning attack will take place over an extended period of time and will manipulate conversations to cause the LLM to produce unsafe or unintended outputs.

As a result, protecting the confidentiality, integrity, and availability of LLM interactions has become a key consideration for enterprises and institutions when deploying LLM technology. Without effective protective measures, LLMs will continue to be susceptible to abuse, adversary-based alterations, and the loss of confidential information.

1.3 Limitations of Existing Systems

Existing filtering systems for LLMs/AI security are mostly built upon singular-layer filtering methodologies lacking contextual awareness. Because filtering is inherently contextually void, these systems are unable to detect attacks occurring over time, attempts to covertly manipulate the system in unknown ways, and adversarial prompts being spread across interactions.

Many of the available tools also only focus on text-based content and provide little to no protection against multi-modal attacks, such as those using audio or image data. Some cutting-edge filtering systems do include a second LLM for validating filtering decisions, but that approach is expensive and slows down the response time of LLMs significantly.

Another major issue of filtering is the practice of binary blocking; that is, blocking a prompt simply because one small portion was malicious, thereby creating a poor user experience and potentially blocking valid user requests. The majority of LLM filtering systems also do not provide explainability or a way to monitor blocks in real-time, making it difficult for administrators to have an understanding of how and why a prompt was either blocked or modified.

1.4 Proposed Approach

To address the issues faced by Many Clients when deploying LLMs, this work proposes an Adaptive Contextual Multi-Layer Prompt Security System Output Governance Framework that will allow for secure deployment of LLMs via a multi-dimensional process (i.e., keyword, semantic, conversational context, and user behaviour) for processing prompts and ingesting into LLMs.

The framework includes a multi-layer prompt inspection framework with contextually aware adversarial detection capabilities for the identification of malicious or suspicious activity across a conversation history. The framework also supports multi-modal input through speech-to-text conversion and optical character recognition (OCR) based extraction techniques for text, audio, and image inputs.

To provide safe and reliable generation of outputs from LLMs, the framework includes governance mechanisms for output generation via policy validation, sensitive data leakage detection, and sandbox verification. The framework is also designed for greater transparency, accountability, and secure management by means of the combination of role based access control and explainable AI Monitoring. This framework is intended to provide clients with an efficient Lightning fast scalable solution for secure interactions with LLMs.

2 PROBLEM STATEMENT

Large Language Models are vulnerable to prompt injection, adversarial manipulation, context poisoning, and sensitive data leakage. Existing safety mechanisms rely mainly on static filtering or basic moderation that fail to detect indirect and context-aware attacks.

Limited by the rapid adoption of LLMs in enterprise systems, virtual assistants, education systems, or AI automation tools, keeping secure interaction with such models has become a significant challenge. The exploitable weaknesses in prompt handling can allow an attacker to manipulate how the model behaves. Bypassing safety constraints, extracting confidential information and causing risk occurs with the integration of LLMs into systems which process sensitive empanelling of user or organizational information.

Another limitation inherent within current defenses is the evaluation of prompts in isolation, without concern to other conversations. Attackers can modify prompts over time with multiple interactions (and do this incrementally). Therefore, traditional forms of filtering will not detect incremental prompt modifications.

As LLMs and other AI increasingly support multi-modal input forms such as images and audio, the attack surface being exploited is growing exponentially more than the text only methods traditional filtering at present. Thus, there is an urgent need for a context aware, multi-level security structure to be established for trustworthiness and secure deployment of LLMs.

3 OBJECTIVES

1. Understanding and detecting prompt patterns of legitimate, malicious and adversarial prompts, and create a multi-layer prompt inspection system with lexical and semantic filtering layers.
2. Develop multi-modal processing (text, audio, image) of the input data stream; develop validation, governance, sandbox and explainability processes for deploying LLM's safely and transparently.
3. Create Contextual Adversarial Detectors that will detect and prevent prompt injections and manipulations by using the past history of inputs to create a more effective overall process for LLM usage.

4 LITERATURE SURVEY

Overview:

The focus of recent research on LLM's has changed from purely demonstrating how to create an automatic attack system to now focusing on developing security mechanisms through Adaptive Security Techniques, Prompt Engineering, and Multi-Agent Systems that can be utilized in defending against the above types of attack strategies. Lastly, multimodal input (i.e. multi-modal input) attacks will continue evolving.

A range of approaches that currently exist do not have the ability to scale, be efficient, or be practical in the real world. The lack of proactive measures to defend against evolving threats indicates a need for scalable and effective security solutions.

4.1 Prompt Injection Attacks & Evaluation Frameworks

Initial research in LLM security focused on understanding and evaluating the vulnerabilities of language models against adversarial prompts. Paper [1] introduced a framework for assessing the robustness of LLMs against prompt injection attacks and provided comparative evaluation metrics for different models. This work mainly focused on measuring vulnerabilities rather than preventing attacks.

Similarly, Paper [2] proposed a semantic and rule-based security framework for identifying malicious instructions within prompts. The framework improved the detection of unsafe prompts but relied heavily on predefined rules and lacked adaptability to evolving attacks.

Paper [7] further expanded this area by generating adversarial prompts using reconstruction-based methods, demonstrating how easily existing safety mechanisms can be bypassed. This highlighted the fragility of alignment techniques currently used in LLM systems.

Insight: The initial exposure of LLM's vulnerabilities and the fragility of their alignment have also been discussed.

Limitations:

- There are few to no dynamic defense mechanisms.
- Evaluation is the primary focus rather than Prevention.
- Lack of multimodal awareness.

4.2 Prompt Injection Defense & Security Frameworks

Research gradually shifted from evaluation toward adaptive defense strategies for prompt injection attacks. Paper [3] introduced automated prompt engineering techniques to improve model alignment and reduce unsafe outputs. While the approach improves prompt safety, it still struggles against unseen attack patterns.

Paper [4] proposed an LLM-based reasoning framework that uses adaptive analysis to identify malicious prompts and adversarial instructions. This improves contextual understanding but introduces higher computational overhead due to continuous reasoning operations.

Similarly, Paper [6] developed a multi-agent defense framework where multiple agents collaboratively analyze prompts and identify threats. This approach enhances detection capability for complex attacks but increases system complexity and synchronization overhead.

Insight: The evolution of defense mechanisms (to include intelligent and adaptive multi-agent systems) is demonstrated.

Limitations:

- Vulnerability to unseen attacks
- High computational overhead
- Lack of standard benchmarks

4.3 Multimodal Prompt Injection & Vision-Language Attacks

Recent research has shown that prompt injection attacks are no longer limited to text-based systems. Paper [5] demonstrated that adversarial commands can be hidden inside image files, which are later extracted and interpreted as instructions by LLM systems.

Similarly, Paper [11] explored hidden visual prompts embedded within images to manipulate model behavior. Papers [12] and [13] further extended this area by demonstrating adversarial cross-model attacks on vision-language systems.

Paper [8] provided a broader overview of multimodal vulnerabilities, showing how modern AI systems become more vulnerable when supporting text, image, and audio inputs together.

Insight: The attack surface of multimodal systems dramatically increases attack surface.

Limitations:

- Limited defense mechanisms
- Lack of standardized datasets
- Poor understanding of multimodal interactions

4.4 Sophisticated & Multi-Agent Attack Strategies

Advanced attack strategies are increasingly becoming more automated and stealth-oriented. Paper [14] discussed multi-agent camouflage attacks where coordinated agents collaborate to bypass existing defense systems through deception techniques.

Similarly, Paper [7] explored reconstruction-based adversarial generation techniques that dynamically generate malicious prompts capable of bypassing static filtering systems.

Insight: The trend is moving towards the Automation of attacks being more Adaptive, Intelligent and Stealthy.

Limitations:

- Defense mechanisms lag behind attacks
- Limited real-world validation
- Focus more on attack than defense

4.5 Jailbreak Attacks through Audio & Speech

Research has also revealed major vulnerabilities in audio-enabled AI systems. Paper [15] demonstrated jailbreak attacks performed through adversarial audio inputs capable of bypassing speech-based interfaces.

Likewise, Paper [16] explored adversarial attacks on SpeechGPT systems using manipulated speech signals to inject malicious instructions. **Insight:** Since Audio Interfaces are unregulated, they provide an attack vector that is currently not being monitored.

Limitations:

- Limited mitigation strategies
- Low research maturity
- Model-dependent designs

4.6 Context-Aware LLMs in Recommendation Systems

Large Language Models are increasingly being used in context-aware recommendation systems. Paper [17] introduced an ontology-based recommendation framework that improves personalization using contextual understanding.

Similarly, Paper [10] proposed sentiment-driven recommendation techniques using LLMs to better understand user preferences and interactions.

Insight: LLMs enhance personalization and context-awareness.

Limitations:

- Limited focus on security
- Challenges in dynamic interaction
- Scalability issues

4.7 LLM-based Evaluation, Governance & Compliance

The use of LLMs for governance and compliance monitoring is also becoming an important research area. Paper [18] introduced SmartJudge, an LLM-based auditing framework for smart contract analysis and validation.

Similarly, Paper [9] explored the use of language models for governance enforcement and personally identifiable information (PII) compliance.

Insight: Language models are evolving into the means for monitoring and auditing systems

Limitations:

- Risk of bias.
- Limited explainability.
- Dependence on model quality.

4.8 Secure & Private LLM Platforms

Research has also focused on secure deployment and privacy-preserving LLM systems. Paper [19] proposed PrivMLLM, a secure computing framework designed to improve privacy and protection during LLM inference.

Similarly, Paper [23] discussed broader security and privacy management issues associated with deploying language-based systems in real-world environments.

Insight: Focus on secure deployment of LLMs. **Limitations:**

- High computational cost
- Limited real-time usage
- Trade-off between performance and security

4.9 Cybersecurity & Adaptive Defense Systems

Large Language Models are increasingly being applied in cybersecurity systems. Paper [21] proposed adaptive security mechanisms for Industrial Internet of Things (IIoT) environments using LLM-based intelligence.

Similarly, Paper [22] explored the use of LLMs for generating firewall policies dynamically based on changing security requirements. **Insight:** Large Language Models are providing Intelligent Automated Solutions for Cybersecurity.

Limitations:

- Limited real-world validation
- Risk of incorrect automated decisions
- Dependence on input quality

4.10 Survey of Jailbreak Attacks

Paper [20] provides a detailed classification of types of jailbreak attacks, including Prompt-based, Role-playing, or Multi-turn; and discusses the Defense-alignment gap associated with these jailbreak attacks.

Insight: Jailbreak attacks are evolving faster than defenses.

Limitations:

- Limited implementation focus
- Lack of practical solutions

Sl. No.	Author & Year	Title / Work	Method / Approach	Key Contribution	Limitation
1	Yip et al. (2023)	Evaluation Framework for Prompt Injection	Evaluation metrics framework	Measures resilience of LLMs	No defense mechanism
2	Zhao et al. (2023)	Security Framework against Prompt Injection	Rule-based + filtering	Detects malicious prompts	Limited adaptability
3	Kwon et al. (2024)	Automatic Prompt Engineering Defense	Prompt engineering	Dynamic defense mechanism	Weak for complex attacks
4	Muliarevych et al. (2024)	LLM-Driven Defense System	AI-based detection	Adaptive security approach	Depends on model accuracy
5	Nagaraja et al. (2026)	Image Prompt Injection Study	Multimodal attack analysis	Exploits image-based prompts	Lack of defenses
6	Gosmar et al. (2025)	Multi-Agent Detection Framework	Multi-agent NLP system	Improves detection accuracy	High complexity
7	Yu et al. (2025)	Reconstruction-Based Prompt Attack	Adversarial prompt generation	Generates effective attacks	No mitigation
8	Liu et al. (2025)	Vision-Language Attack Survey	Survey + taxonomy	Categorizes multimodal threats	No implementation
9	Mavani et al. (2026)	AI Governance Framework	Monitoring + compliance	Ensures PII protection	Complex system
10	Ahuja et al. (2025)	LLM-based Recommendation	Sentiment + context-aware	Improves personalization	Ignores security
11	Zhang et al. (2025)	Image-based Prompt Injection	Visual adversarial inputs	Hijacks multimodal LLMs	Weak defense
12	Vo et al. (2024)	Adversarial Captioning Attack	Prompt engineering	Breaks captioning models	Limited scope
13	Peng et al. (2025)	Unified Prompt Attack (UPAM)	Gradient-based optimization	Efficient cross-model attack	Requires queries
14	Li et al. (2026)	Camouflage Attack Framework	Multi-agent system	Stealthy adversarial prompts	Complex coordination
15	Chen et al. (2026)	AUDIOJAILBREAK	Audio adversarial attack	Exploits speech models	Limited defenses
16	Ma et al. (2025)	SpeechGPT Jailbreak Attack	White-box token attack	High success rate (audio)	Model-specific
17	Gao et al. (2017)	Context-Aware Recommendation	Ontology-based system	Improves personalization	Not scalable
18	Mothukuri et al. (2025)	Smart Contract Auditor Evaluation	LLM-as-judge	Evaluates AI systems	Reliability issues
19	Xue et al. (2025)	PrivMLLM	Secure multi-party computation	Privacy-preserving inference	High overhead
20	Knowlton et al. (2026)	Jailbreak Attack Survey	Survey + taxonomy	Defines attack categories	No implementation
21	Mao et al. (2026)	Cognitive Security for IIoT	LLM + RAG framework	Adaptive cybersecurity	Needs real-world testing
22	Rahmani et al. (2025)	FireAI Framework	Generative AI firewall	Automated policy generation	Risk of errors
23	Rathod et al. (2025)	LLM Security Challenges	Survey + analysis	Identifies risks & solutions	No practical solution

Table 1: Literature Review Summary

5 RESEARCH GAP AND RESEARCH CHALLENGES

Research Gaps Identified

- Most existing studies tend to explore the effectiveness of individual prompts without incorporating the conversational elements. Practically, these methods fail when dealing with multi-turn or ”slowburn” attacks, where the malicious intent is gradually revealed over a series of interactions.
- There is still a lack of adequate multimodal security. At the present, the solutions available only analyze text, image, or audio separately, but they do not offer cross-modal detection, leaving attack avenues open through subtle input channels.
- A lot of the methods are based on a single layer or heavyweight, LLM-driven defensive mechanisms. This not only raises the response times but also mostly focuses on the blocking of output instead of reconstructing secure intent, which disturbs the overall user experience.

Research Challenges

- Real-time and context-aware detection remains difficult mainly because tracking multi-turn conversations and grasping semantic changes is complex.
- Multimodal security adds another layer of difficulty since different data types (text audio image) not only need their own processing methods but also the overall defense strategy should remain coherent.
- It is quite a challenge to devise a multi-layer security framework that not only meets the requirements of speed, accuracy, and scalability but also integrates secure output reconstruction mechanisms.

6 PROPOSED METHODOLOGY

6.1 Approach

The Adaptive Context-Aware Multi-Layer Prompt Security and Output Governance Framework is a promising security concept that protects interaction with Large Language Models (LLMs) in a hybrid layered architectural style.Pipelines are utilized by the system to secure both inputs and outputs from prompt injections, adversarial attacks and accidental disclosure of secure information. The system incorporates the three tenets of multi-layered analyzing prompts; discovering adversarial contextual elements; securing outputs from being misused. The first step is filtering prompts; both procedurally through lexical and semantical means, to identify any possible threats prior to their being deployed. The system conducts ongoing reviews of communication through tools and identifying patterns of abnormal behavior, deviations from expected standards; and identifying instances of minor manipulation(s) following the initial deployment of the prompts.

In order to process the multi-modal inputs of the inputs, the system enables the pre-processing of text, audio and visual elements. For example, audio inputs will be converted to written (text) language via speech to text technologies, and visual inputs will be recognized via optical character recognition (OCR) technologies; as a result, all input sources will ultimately be converted to a text medium enabling uniformity of processing by the prompts.

In respect to outputs, governance such as whether sensitive information may have been leaked; whether a response has been or otherwise unsafe; whether a breach has occurred. For programming, execution will be safeguarded using sandboxing methodologies that will also mitigate any risk posed by malicious intent.

In addition, we manage user action by way of abusing access to our system or recording inappropriate activity by users through an abuse detection process. Within our monitoring pane, you will find a separate tab that displays detailed log files so that we can describe how we arrived at these decisions and save the history of system usage for review later.

Overall, the proposed approach employs multiple modules in a pipeline, such as prompt analysis, contextual analysis, multimodal preprocessing, output validation and monitoring, so that they provide a secure way to use LLM-based systems at scale with reliability.

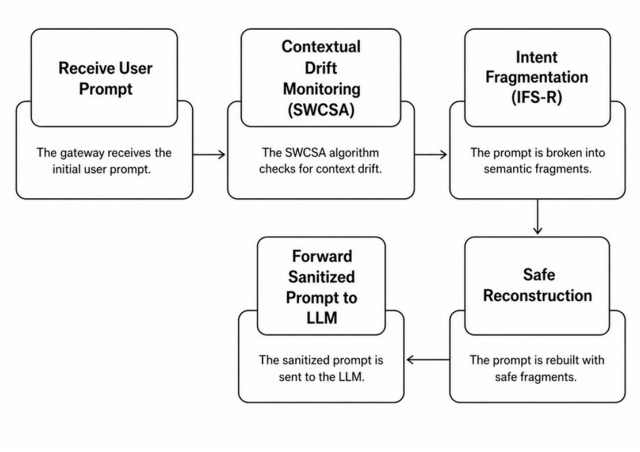


Figure 1: Context-Aware Prompt Processing and Security Pipeline

6.2 System Architecture

Our system architecture is a layered security architecture, with each layer serving a specific purpose to interact safely and securely with Large Language Models (LLMs). The system takes inputs, interprets context, identifies risks and verifies the outputs in a step-wise fashion.

6.2.1 Multi-Modal Input Layer

This is the starting point of the system, and receives inputs in various forms such as text, audio and images. It translates the audio into text using speech-to-text, and image into text using OCR and metadata extraction. Moreover, signal sanitization methods such as frequency filtering are employed to eliminate covert or malicious signals.

6.2.2 Text Normalization & Preprocessing Layer

The standardization and textual representation of inputs occur in this layer through a pre-processing gateway. It cleans, tokenizes and normalises input to maintain uniformity for analysis.

6.2.3 Context Buffer Layer (Sliding Window)

This uses sliding window buffer to keep track of recent interactions. It retains the recent messages for context-sensitive analysis and conversation monitoring.

6.2.4 Sliding Window Context Shift Analyzer (SWCSA)

The Sliding Window Context Shift Analyzer tracks context shifts in the conversation. It measures unusual context drift, role shifting, and topic drift through embeddings, pattern matching and entropy.

6.2.5 Drift Score Calculation Layer

This layer is responsible for computing a drift score. If this score is above a certain threshold, the prompt is marked for investigation.

6.2.6 Intent Fragmentation & Reconstruction layer (IFS-R)

It divides prompts into smaller sub-intents to detect harmful sub-intents. Risky elements are replaced or deleted, and the prompt is safely reconstructed.

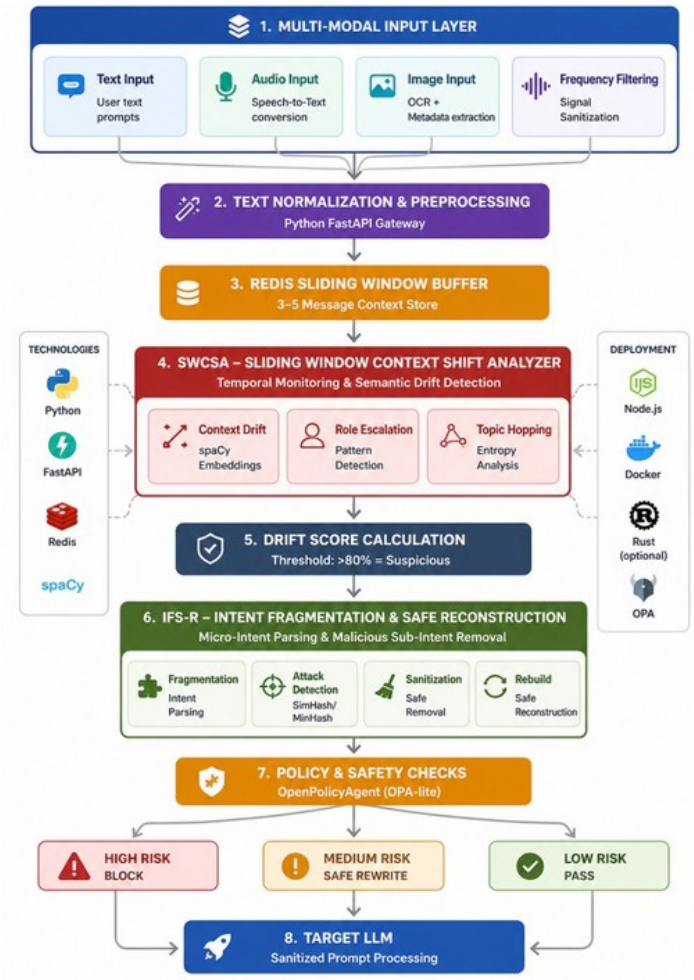


Figure 2: System Architecture of Adaptive Multi-Layer Prompt Security Framework

6.2.7 Policy and Safety Enforcement Layer

This layer uses rule engines to make policy-based checks and decide a level of risk. The prompt can be blocked, rewritten or passed through, depending on the risk level.

6.2.8 Target LLM Processing Layer

Only safe and filtered prompts are sent to the LLM. This guarantees safe and compliant prompting, avoiding misuse and misbehaviour.

6.2.9 Supporting Components

In addition to these features, this platform offers role-based access control (RBAC) to manage user permissions and roles, as well as real-time detection of abuse or any immediate threats to people or property; an Explainable AI Dashboard which provides full transparency in terms of how decisions have been made by the system and how the models behave; and finally, development continues on lightweight technologies capable of faster development and easier scalability.

6.3 Tools and Technologies

To build the suggested solution, we will employ the following instrumentation and technology stack:

Technological Stack:

Core Engine-

- Python FastAPI / Node.js (Microsecond API gateway)
- Rust Micro Service (Optional:For lightning fast text processing)
- Redis 3-5 message sliding buffer
- A Rule based Semantic Micro-Parser to provide intent fragmentation
- Local CPU Embedding Lite (optional) for semantic drift scoring
- No-GPU (real-time processing)

Algorithms-

- SWCSA (Sliding Window Context Shift Analyzer)
  Measures semantic drift entropy drift of time
- IFS-R (Intent Fragmentation + Safe Reconstruction)
  Micro-Intent graph parsing safety map replay

Deployment-

- Docker + Kubernetes (optional) ready
- Edge / Browser / On-device LLM gateways

Dependencies-

- spaCy / Tokenizers - Very fast CPU only Tokenization
- SimHash / MinHash - detects similar or sneaky rewrites
- Small Sentence Embedding Model (30MB) - optional for semantic drift
- OpenPolicyAgent (OPA-lite) - policy checks for unsafe micro-intents
- No GPU frameworks

Note: Full pipeline can run on CPU with ¡5 ms delay

Hardware Requirements-

- Minimum 8GB RAM
- CPU processing only (does not require GPU processing but will enhance when using advanced models).

6.4 Process

The framework for the Multi-Layer Prompt Security and Output Governance is envisioned as a pipeline that processes the input that comes from the user, detects threats, and reacts securely with the Large Language Model (LLM). Each section of the pipeline serves to either identify, block, or mitigate the threat posed by an injection of a prompt to the LLM while ensuring the system is functioning efficiently and reliably.

6.4.1 Data Collection, Preprocessing and Multi-Modal Processing

As the first stage of this workflow, the collection of different types of data, such as user-generated prompts, malicious prompts, and examples of prompt injection patterns (found in both research papers and testing platforms), will allow us to identify different attack possibilities as well as train the detection system.

After collection, the inputs will be pre-processed. For text prompts received from the user, they will be cleaned and standardised; speech inputs will be converted to text; and image inputs will be analysed by OCR and extracted for their metadata. As a result, all the inputs will be standardised prior to processing.

This system can accept audio and images as well as text inputs. To eliminate hidden adversarial content from those sources, pre-processing methods such as audio filtering and image denoising are employed.

6.4.2 Prompt Inspection and Context-Aware Analysis

The sanitized input will undergo multiple stages of analysis, including both lexical and semantic filtering, to help determine the presence of potentially malicious language, unsafe commands, and/or known attack signatures.

The system maintains a record of a small number of recent conversations, and using the Sliding Window Context Shift Analyzer (SWCSA), looks at how the flow of dialogue is happening. During this time, it will find semantic drift, role escalation, or any other type of unusual behavior occurring over various turns of the conversation.

This process generates a drift score for the prompt based on the detection of changes to the semantic meaning within the conversation. When drift scores exceed predefined limits, the originator's prompt may be considered as a possibility for having malicious intent and require further investigation.

6.4.3 Intent Reconstruction and Policy Enforcement

It is related to detecting malicious intents. Each prompt is separated into smaller parts (sub-intent) and each part is checked for harm to someone. Each of those parts can be changed or eliminated and the prompt will be converted into a form that will not change the user's original intent.

The cleaned prompt will be validated against a security policy to determine whether it should be blocked, rewritten or allowed to pass depending on the risk rating assigned to each prompt. The security policy for the prompt will use rule base engines to determine if the prompt can be blocked, rewritten or allowed to pass based on the level of risk.

6.4.4 Post-Generation Governance and Monitoring

The LLM generated content will be reviewed for having any sensitive information, unsafe instructions or violations of policy. Where code is created, sandboxing is used to safely execute the code.

The Explainable AI component will document the rationale for all decisions made about blocking, modifying or accepting prompts in the monitoring and explainability components of the system to allow the System Administrator to monitor the security and functionality of the system.

6.4.5 Evaluation Metrics

It is tested on metrics including accuracy, false positives, response time, and prevention of prompt injection attacks and unsafe outputs.

6.4.6 Technical and Economic Feasibility

You can implement the project with Python, machine learning libraries, and large language model (LLM) APIs. You'll also need modules for natural language processing, speech-to-text, and optical character recognition. On the server side, one of many frameworks can be chosen. This kind of application can be run on an ordinary computer that has a CPU. In summary, the project is doable from a technical point of view since the necessary technologies and open-source frameworks are available.

Mostly opensource technologies are used Less hardware is required No special infrastructure needed Development can be done with existing institutional resources Hence the project is financially feasible for academic implementation

7 HIGHLIGHTS FROM PHASE 1 REVIEW 1

At the first phase review, the panel of experts suggested that the project should be made more brief and well-ordered, and the objectives of the project could be slightly modified. They pointed out that the project had, at the first phase, only identified the problem and proposed a multi-layer security architecture. Also, the panel mentioned that the system mustn't be text-only but should incorporate multi-modal features.

Feedback on the design changed the way the architecture was presented and simplified the processing path of prompts through all security layers. The goal of the project has evolved to provide more emphasis on the primary functions of the overall system (multi-layered analysis of prompts, context sensitive detection of adversarial prompts, and controlling the output of a prompt). The design was modified to provide for the use of multiple modes of input by allowing for the integration of images and audio into the text input. This was accomplished through OCR and metadata extraction for images, as well as speech-to-text for audio, where all forms of input will undergo the same type of security validation during processing.io, with all formats of input undergoing similar security validation.

These enhancements increase the system's functionality and suitability for practical applications. The project now effectively presents a simpler, more focused, and more realistic approach to secure interaction with Large Language Models (LLMs), and meets the requirements highlighted by the review panel.

8 EXPECTED OUTCOMES

- The new system will offer a safe and trustworthy way to interact with Large Language Models (LLMs) by defending against prompt injection and other attacks.
- It will effectively identify and neutralise malicious prompts, including indirect and multi-step attacks, by considering both prompt and context. It will avoid false positives by modifying unsafe prompts to be safe and avoid blocking them, enhancing user experience.
- The system will support multi-modal processing (text, audio, image), allowing it to process text, audio and image inputs in the same way, making it possible to detect and mitigate malicious prompts in different formats.
- The use of output governance will ensure that LLM responses will be safe, policy-compliant and won't leak sensitive information. This will also include controlled execution which uses sandbox validation for code.
- Admins will have access to live monitoring tools, be able to validate their decisions through traceability, and gain greater control and transparency over their actions. The authors of the paper demonstrate a minimalist methodology that has a low latency and can scale with both the needs of the business and the nature of the world.

Overall, the proposed effort will provide a holistic, efficient, and pragmatic solution for safely deploying large-scale language models.

9 CONCLUSION

This research project proposes Adaptive Context-Aware Multi-Layer Prompt Security and Output Governance Framework that can address security issues of Large Language Models (LLMs) such as prompt injections, manipulations, and leakages. Besides, it highlights the limitation of multi-layer prompt inspection, context-aware analysis, multi-modal input processing and output governance to solution of security without filter-based only.

Through context and pattern analysis, the framework enhances not only the identification of direct and indirect attacks but also the generation of safe prompts instead of input-filtering in order to secure the user experience.The integration of text, audio, and visual processing will ensure that security is maintained regardless of the input method, while sandboxing and policy enforcement will offer additional security for the resulting model.

This framework provides a lightweight, scalable, and explainable approach to deploying large-scale language models, making the framework feasible in both real-world and enterprise context. Besides exhibiting the practical usage of secure interactions with LLMs, it also aims to provide the theoretical foundation for AI safety and governance based upon this work.

10 REFERENCES

[1] D. W. Yip *et al.*, "A Novel Evaluation Framework for Assessing Resilience Against Prompt Injection Attacks in Large Language Models," 2023.

[2] Y. Zhao *et al.*, "A Novel Security Framework against Prompt Injection Attacks," 2023.

[3] H. Kwon *et al.*, "Defending Against Prompt Injection Attacks Using Automatic System Prompt Engineering," 2024.

[4] O. Muliarevych *et al.*, "Enhancing System Security: LLM-Driven Defense against Prompt Injection Vulnerabilities," 2024.

[5] N. Nagaraja *et al.*, "From Pixels to Prompts: A Systematic Study and Introduction to Image Prompt Injection Attacks," 2026.

[6] D. Gosmar *et al.*, "Prompt Injection Detection and Mitigation with AI Multiagent NLP-Based Agentic Frameworks," 2025.

[7] Z. Yu *et al.*, "Reconstruction-Based Prompt Generation Algorithm for Prompt Injection Attacks," 2025.

[8] D. Liu *et al.*, "A Survey of Attacks on Large Vision-Language Models: Resources, Advances, and Future Trends," 2025.

[9] V. K. Mavani *et al.*, "An Autonomous Governance Framework for Generative AI: Real-Time PII Redaction and Compliance in LLM-Driven Data Pipelines," 2026.

[10] A. Ahuja *et al.*, "Context-Aware Product Recommendations for Emerging Markets: A Sentiment-Driven LLM Approach," 2025.

[11] L. Zhang *et al.*, "Image-Based Prompt Injection: Hijacking Multimodal LLMs through Visually Embedded Adversarial Instructions," 2025.

[12] H. Vo *et al.*, "Prompt Engineering Adversarial Attack against Image Captioning Models," 2024.

[13] D. Peng *et al.*, "Unified Prompt Attack Against Text-to-Image Generation Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025.

[14] X. Li *et al.*, "A Synergistic Multi-Agent Framework for Camouflage Attack on Large Language Models," *IEEE Trans. Audio, Speech Lang. Process.*, 2026.

[15] G. Chen *et al.*, "AUDIOJAILBREAK: Jailbreak Attacks against End-to-End Large Audio-Language Models," 2026.

[16] B. Ma *et al.*, "Audio Jailbreak Attacks: Exposing Vulnerabilities in SpeechGPT in a White-Box Framework," 2025.

[17] Q. Gao *et al.*, "A Conditional Context-Awareness Ontology-Based Personalized Recommendation Approach for E-Reading," 2017.

[18] V. Mothukuri *et al.*, "Automated Judging of LLM-Based Smart Contract Security Auditors," 2025.

[19] Y. Xue *et al.*, "PrivMLLM: Efficient Three-Party Multimodal Large Language Model Secure Inference Supported Prompt Privacy," 2025.

[20] B. Knowlton *et al.*, "Prompt-Based Jailbreaking of Leading LLM Chatbots: A Survey of Attacks and Defenses," 2026.

[21] J. Mao *et al.*, "Cognitive Security for AI-Native IIoT: An LLM-Centric Adaptive Defense Framework," 2026.

[22] A. Rahmani *et al.*, "FireAI: A Framework for Dynamic Firewall Policy Generation using Generative Artificial Intelligence," 2025.

[23] V. Rathod *et al.*, "Privacy and Security Challenges in Large Language Models," 2025.